Mathematical modelling of shallow water flows with application to Moreton Bay, Brisbane

Mathematical modelling of shallow water flows with application to Moreton Bay, Brisbane

Clare Louise Bailey

Department of Civil and Building Engineering

Loughborough University

A Thesis Submitted for the Degree of

Doctor of Philosophy

May 2010

Abstract

A finite volume, shock-capturing scheme is used to solve the shallow water equations on unstructured triangular meshes. The conditions are characterised by: slow flow velocities (up to $1\,\mathrm{ms}^{-1}$), long time scale (around 10 days), and large domains (50-100 km across). Systematic verification is carried out by comparing numerical with analytical results, and by comparing parameter variation in the numerical scheme with perturbation analysis, and good agreement is found. It is the first time a shock-capturing scheme has been applied to slow flows in Moreton Bay.

The scheme is used to simulate transport of a pollutant in Moreton Bay, to the east of the city of Brisbane, Australia. Tidal effects are simulated using a sinusoidal time-dependent boundary condition. An advection equation is solved to model the path of a contaminant that is released in the bay, and the effect of tide and wind on the contaminant is studied. Calibration is done by comparing numerical results with measurements made at a study site in Moreton Bay.

It is found that variation in the wind speed and bed friction coefficients changes the solution in the way predicted by the asymptotics. These results vary according to the shape of the bathymetry of the domain: in shallower areas, flow is more subject to shear and hence changes in wind speed or bed friction had a greater effect in adding energy to the system.

The results also show that the time-dependent boundary condition reproduces the tidal effects that are found on the Queensland coast, *i.e.* semi-diurnal with amplitude of about 1 metre, to a reasonable degree. It is also found that the simulated path of a pollutant agrees with field measurements. The computer model means different wind speeds and directions can be tested which allows management decisions to be made about which conditions have the least damaging effect on the area.

Keywords

Shallow water equations, pollutant transport, environment, FVM, Java, unstructured triangular mesh, Moreton Bay.

Acknowledgements

I would like to thank my supervisors Graham Sander and Cecil Scott of Loughborough University and John Norbury of the University of Oxford for many valuable and interesting discussions. I would also like to thank the Department of Civil and Building Engineering for funding me for the three years of this thesis.

I am particularly grateful to my Postgraduate Research Administrator, Helen Newbold, who has always been friendly and welcoming, and who has dealt with my questions and issues efficiently. The IT staff John Salisbury and Stephen Grieve have solved my computing issues, for which I thank them. Peter Heng gave me a lot of help with programming, which I am extremely grateful for.

The first two years of my PhD were spent at the Oxford Institute for Applied and Industrial Mathematics (OCIAM) as a visiting student. For the warm welcome I received, I would like to thank John Ockendon, Sam Howison and Chris Breward. Working at OCIAM has helped me develop in both professional and academic ways, and it has been a very great privilege.

In addition I would like to thank Andrew Fowler for the inspiring seminars and workshops on geoscientific mathematics, and all the postgrads including Rebecca Shipley, Rosemary Dyson and Chris Bell, who offered support and encouragement in kind and generous ways.

During the last two years, I worked full time while studying in my free time. My employers have shown support for me during this time. Without their help it would have been considerably more difficult to finish writing up, and I acknowledge them gratefully.

Finally, grateful acknowledgement for the support in all forms must be given to my parents, sister, godparents and most excellent boyfriend.

May 2009

Clare Bailey

Contents

List of Figures

CHAPTER 1

Introduction

Mathematical modelling is an attempt to simulate the real world. The mathematical model provides insight into a problem where observed data are incomplete, field studies are impractical or costly, or predictions based on particular parameters are required. The successful model will fit the existing observed data, but will also be able to extrapolate beyond situations originally described. It has applications in many areas including economics, financial markets, biology, physics, engineering, and environmental sciences.

One form a model can take is a set of one or more partial differential equations (PDEs). In this thesis we consider the two-dimensional shallow water equations (2D SWEs), a well-known hyperbolic system to which analytical solutions are usually only available under certain restrictive assumptions. In the past, modellers could take either an analytical or theoretical approach or an experimental one. The former is limited by the power of known analytical techniques, and even a simple system may have no analytical solutions. On the other hand the experimental approach can be costly, requires expertise in using specialist instruments, and may not scale up reliably to full size. As a result, numerical techniques have become more popular as increased computing power means parameter values can be easily adjusted, complex geometries can be handled, and complex systems can be treated in a reasonable computing time.

The solutions that a numerical scheme produces must be realistic and physically relevant, despite any simplifications of physical reality in the design of the code. There are generally two

ways of ensuring that the system meets the design criteria:

- Simulations are compared with analytical results. Although analytical results may require a high degree of simplification, they provide a means of testing that the code satisfies the specified demands. This stage is known as *validation*. It is the degree to which the simulation is an accurate model of the real world – we built the right model.

- Model parameters are adjusted so that the simulated most closely matches the observed. This is known as *calibration* or *verification*. We determine that the model matches the developer's descriptions and specifications, or in this case, the physical reality – we built the model right (see for example Labrosse *et al.* (2007)).

Successful validation and calibration means the model can be applied to other domains and the results will be reliable and accurate. The processes connect the physical and mathematical (or, real and numerical) worlds by ensuring that one matches the other.

1.1 Motivation

The city of Brisbane and much of the surrounding area acts as a source of agricultural and industrial pollutant to the Brisbane river, either during periods of rainfall in which rain washes pesticides and fertilisers off the land and into the river system, or by direct discharge of industrial effluent into the river. The Brisbane river then discharges into Moreton Bay, lying to the east of the city. Ideally, the pollutant is then washed out to the open sea, but Moreton Bay is partially enclosed, meaning it is possible that the pollutant remains in the Bay for some time.

Management of catchment-derived inputs of pollutants is essential to mitigate against detrimental effects on the Moreton Bay ecosystem (Hodge *et al.* (2005)). Moreton Bay is home to a large number of plant, fish, bird and mammal species. Increased levels of nitrogen and phosphorous in the water due to human activity threatens these species' habitats. Excess nutrients in waterways can stimulate the growth of algae to nuisance proportions, diminishing light availability to benthic species (`healthywaterways.org`).

In Queensland, it has been recognised that there is a need to mitigate the risk of catchment-derived inputs of nutrients, sediments and toxicants detrimentally affecting the integrity of the

iconic Great Barrier Reef and Moreton Bay ecosystems (Hodge *et al.* (2005)). A number of state and charitable organisations such as the State Environmental Protection Agency, the Pumicestone Region Catchment Coordination Association, the Australian Marine Conservation Society provide funding for and carry out research into monitoring and assessing geology, hydrology, sediment character and land use patterns.

While monitoring programmes can be valuable, a model that predicts water quality or the transport of a pollutant provides insight into the likely outcomes of various management practices. A particle-tracking model, as used by Periáñez (2004), or an advective model as used in this thesis, is useful as a predictive tool that can be used to assess contamination under different management or meteorological conditions. Many simulations can be made easily by changing parameter values and repeating tests under different conditions.

The shallow water equations (SWEs) have been used to describe mathematically the flow in any area where depth of water is sufficiently small with respect to the horizontal length. It is a classical model based on the Navier-Stokes equations under the assumption of hydrostatic pressure. The few analytical solutions to these equations that exist are useful for code validation. We use a shock-capturing numerical scheme based on the finite volume method (FVM), with an unstructured, triangular grid that has been designed using an object-oriented approach in Java known as *Riemann2D* . It is the first time to our knowledge that a shock-capturing scheme has been applied to Moreton Bay for slow flows, which is important as the scheme is inherently mass-conserving.

1.2 Aims and objectives

A shock-capturing shallow water solver known as *Riemann2D* has been developed by Jha (2006) and others. It is tested under various wind and tide conditions for long problem times (around 10 days) and slow flows, over varying bathymetry.

The main aims of this thesis are:

- to develop a range of perturbation results for low Froude number with corresponding solutions for each order

- to systematically validate the *Riemann2D* code across a range of flow conditions, and against steady flows in idealised circular- and elliptical-shaped domains
- to show that the numerical code produces the correct physical behaviour in modelling the effects of bed friction, wind speed and Coriolis effect
- to extend the code by including time-dependent boundary conditions for *Riemann2D* and show that it displays the correct behaviour in two idealised domains (a square and a circular basin)
- to apply *Riemann2D* to simulate the flow of water and contaminants in a real bay, in particular Moreton Bay on the central east coast of Australia.

1.3 Thesis structure

Chapter 2 contains a literature review of the field. Since this thesis is multi-disciplinary, this chapter aims to provide an outline of the most important and relevant aspects of the three areas of applied mathematics, environmental engineering, and computer science. Some field studies of Moreton Bay are also cited.

Chapter 3 covers some classical analytical techniques for treating nonlinear hyperbolic PDEs, such as linearisation, and finding Riemann invariants. Asymptotic analysis is carried out and solutions given for the case of low Froude number (low velocity) flows, in the cases with and without source terms. These results allow for verification of results in later chapters. The asymptotic solution for stratified shallow water flow is also given.

Chapter 4 describes the design of a general finite volume scheme in 2D, and provides detail about the structure and functionality of *Riemann2D* . The flexibility of a finite volume formulation in dealing with irregular boundaries is demonstrated. *Riemann2D* itself is written using an object-oriented approach in the Java programming language and some of the properties of this language are described. We introduce the design of the time-dependent boundary condition, and the treatment of modelling a passive pollutant, as well as some of the other aspects of *Riemann2D* such as data reconstruction, limiters and source terms.

Chapters 5 and 6 use two simple two-dimensional domains as case studies of *Riemann2D*'s ability to solve the shallow water equations accurately and reliably. In Chapter 5, parameter values are varied and the numerical solutions are compared with the asymptotic results from Chapter 3, and shown to demonstrate expected behaviour. Two known analytical solutions are reproduced numerically as part of the validation stage, and good agreement is found. In Chapter 6 the time-dependent boundary condition is tested on the two simple 2D domains and found to reproduce the expected results. The effect of wind on the path of a passive pollutant in the fluid is measured to compare the relative effects of bathymetry, Coriolis force and tide.

In Chapter 7, the *Riemann2D* model is applied to Moreton Bay. In particular we are interested in the circulation and distribution of a pollutant during several tidal cycles within the Bay. The main source of pollution is the Brisbane river (Wallbrink (2004)), which discharges into the northern end of the Bay. The presence of wind and tidal forcing is shown to have a strong effect on the transport of the pollutant.

Finally, Chapter 8 presents the conclusions and summarises the research carried out in this thesis, as well as proposing potential future research directions.

CHAPTER 2

Literature review

This chapter provides background to the work of this thesis, in outlining the main areas of study. The first section introduces the SWEs and how these form the model to be solved. The second section is an overview of numerical methods: this field has a vast literature so this section aims to provide a brief explanation of what methods are available and to justify our choice of method. The third section is a review of examples where a shallow water numerical scheme has been applied to a real bay and includes examples of pollutant transport models. The fourth section discusses Moreton Bay, some of the ecological challenges facing it and some of the studies done on it. In the fifth section, the basic principles of Object Oriented Programming (OOP) are introduced with some examples of how OOP has been used to model shallow water flow.

2.1 Shallow water equations

Shallow water is defined in the mathematical sense as having a depth that is at most one tenth of the width of the domain it occupies. In our model, Moreton Bay has a planform area of $1\,200\,\mathrm{km}^2$ and an average depth of 10-20 m, which means the aspect ratio is small and the shallow water equations are valid. Under the assumption of hydrostatic pressure, the Navier-Stokes equations can be shown to reduce to the shallow water equations, which can be used to describe the flows in such domains.

Consider the physical configuration of the problem to be as follows. In a bay of sea water (*i.e.* incompressible, homogeneous fluid at constant temperature [1]), take the (x, y) horizontal plane as being parallel to the surface of the still water, and the depth of the water at a given point as $h = h(x, y, t) \geqslant 0$. We denote depth-averaged velocity in the x-direction as $u = u(x, y, t)$ and the depth-averaged velocity in the y-direction as $v = v(x, y, t)$. While the plane $\{z = 0\}$ can be chosen arbitrarily, it is usually positioned at mean sea level. Measuring down from this plane, the (fixed) bottom of the harbour is at depth $z = -b(x, y)$. The equation $z = -b(x, y)$ is the equation for the bottom surface, also known as the *bottom topography* or *bathymetry*, the depth of which is usually assumed to vary with x and y, and in the case of an erodible bed, b varies in time as well. Consider a perturbation in the water's surface of amplitude $\eta(x, y, t)$. So $z = \eta(x, y, t)$ is the instantaneous position of the actual water surface measured from the plane $\{z = 0\}$, and the total depth of the water in general is $h = \eta - b$ (see Figure 2.1).

The governing equations of the motion of water in a shallow bay are usually taken to be the shallow water equations:

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = s_1 \tag{2.1a}$$

$$\frac{\partial (hu)}{\partial t} + \frac{\partial (hu^2 + gh^2/2)}{\partial x} + \frac{\partial (huv)}{\partial y} = s_2 \tag{2.1b}$$

$$\frac{\partial (hv)}{\partial t} + \frac{\partial (huv)}{\partial x} + \frac{\partial (hv^2 + gh^2/2)}{\partial y} = s_3, \tag{2.1c}$$

where g represents acceleration due to gravity, positive measured in the $z < 0$ direction, and (s_1, s_2, s_3) are the source terms as given below.

The first equation represents conservation of water mass, and the second and third represent conservation of momentum in the x- and y-directions respectively. The shallow water model suppresses any vertical motion. The derivation of this model is carried out in detail in Appendix A, and can also be found in the classical texts (see for example Stoker (1957), Whitham (1974) or Ockendon *et al.* (1999)).

The equations above can be written in vector form as

$$\mathbf{q}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{s}, \tag{2.2}$$

1. Temperature is not considered to play any role at all in the flow of this water.

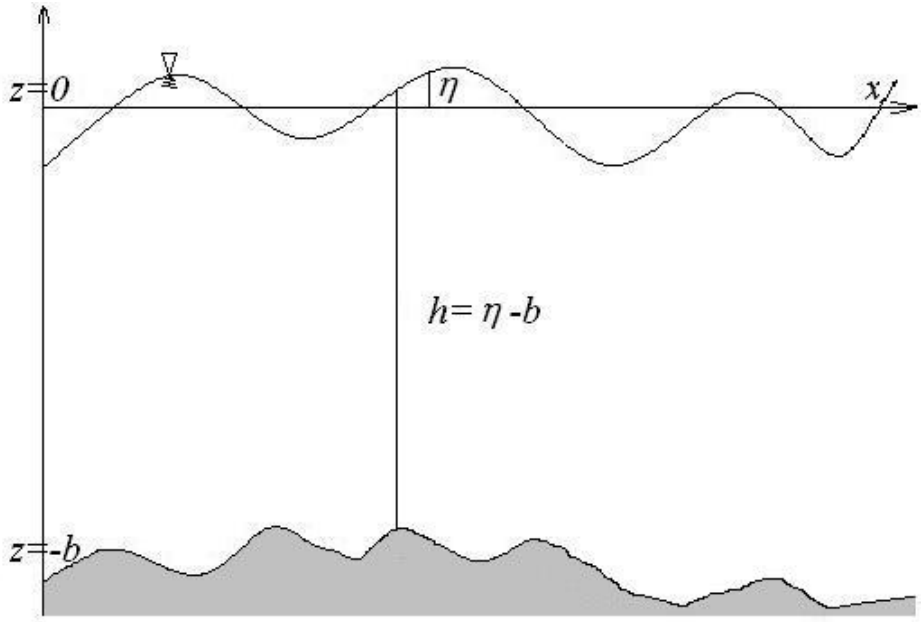


Figure 2.1: Schematic to show geometry of a generic bay. The irregular bed of the bay (shaded grey) is described by the equation $z = -b$ (where b is assumed at least continuously differentiable), the total depth of the water is $h = \eta - b$, and the level of still water (mean sea level) is at $z = 0$. The y-axis is coming out of the page.

where

$$\mathbf{q} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} hu \\ hu^2 + gh^2/2 \\ huv \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} hv \\ huv \\ hv^2 + gh^2/2 \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 0 \\ -ghb_x \\ -ghb_y \end{bmatrix},$$

and where the subscripts t, x, y denote differentiation with respect to that variable. In vector form, the unknown is $\mathbf{q} = [h,\, hu,\, hv]^{\mathrm{T}}$ (the vector of conserved variables), and $\mathbf{F}_x$, $\mathbf{G}_y$ are the divergence of the x, y flux vectors respectively. The right hand side vector of source terms, $\mathbf{s}$, contains the effect of bottom topography. Other source terms may be included such as the effects of wind stress, bed friction and Coriolis force and these are discussed in the next section.

Equation (2.2) is written in conserved form, but can alternatively be written in terms of the non-conserved variables (h, u, v) as

$$h_t + \nabla \cdot (h\mathbf{u}) = 0 \tag{2.3a}$$

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} + g\nabla h = \hat{\mathbf{s}}, \tag{2.3b}$$

where $\hat{\mathbf{s}}$ is the vector containing the second two terms of $\mathbf{s}$. Note that the preferred choice for formulating a numerical scheme is the conserved form given by (2.2), because it means the scheme is inherently conservative as physically meaningful quantities are used throughout.

2.1.1 Source terms

The terms in the right hand vector of (2.2) represent the effect of the shape of the bottom of the bay on the flow and is the most common choice for the source terms. There are however other forces that affect the flow in the bay, including bed friction, Coriolis force and wind stress. The vector of the right hand side can be generalised to include these forces through

$$(s_2,\, s_3) = -gh\nabla b - gh\mathbf{S_f} + fh\mathbf{k} \wedge \mathbf{u} + \tau, \tag{2.4}$$

where $\mathbf{S_f} = (S_{fx},\, S_{fy})$ represents horizontal bed friction, f is the Coriolis parameter, and $\tau = (\tau_x,\, \tau_y)$ represents horizontal wind stress on the surface of the water.

A source or sink such as rainfall, river discharge, outflow, or evaporation is modelled by the inclusion of a non-zero term on the right hand side of (2.1a). In this work however, we always consider that $s_1 = 0$.

Bed friction is modelled by

$$\mathbf{S_f} = c_f \mathbf{u}|\mathbf{u}|,$$

where $c_f = n^2 h^{-4/3}$ is the bed friction coefficient, and n is Manning's roughness coefficient, or simply 'Manning's n'. Typical values of n are $0.01\,\mathrm{m}^{-1/3}\mathrm{s}$ for a general bed of sand or small stones (Chow (1959), Baines (1974), and Fowler (1997)), and for very rough surfaces such as a floodplain of forest or heavy brush, may be no higher than about $0.1\,\mathrm{m}^{-1/3}\mathrm{s}$.

Alternatively, Chézy's coefficient C may be employed, so that $c_f = h^{-1}C^{-2}$, where C has a typical value of 37-40 $\mathrm{m}^{1/2}\mathrm{s}^{-1}$ (Liang *et al.* (2006), Borthwick *et al.* (2001a)). The Chézy form has the advantage over the Manning's form as there is then no singularity in (2.4) as the depth of the water, h, tends to zero.

Coriolis force is force exerted on a body when it moves in a rotating reference frame. Large atmospheric bodies such as clouds do not move directly from areas of high to low pressure,

but because the Earth is rotating, are pulled to the east (northern hemisphere) or west (southern hemisphere). The expression for Coriolis force is

$$fh\mathbf{k} \wedge \mathbf{u} = (0,\, -fhv,\, fhu),$$

where $f = 2\Omega \sin\varphi$ (in units of s^{-1}) is the Coriolis parameter, Ω the angular velocity of the Earth, φ the latitude. The latitude of Moreton Bay is 27°S, so the Coriolis parameter, f, in this case is $-6.6 \times 10^{-5}\,\mathrm{s}^{-1}$.

Wind The effect on the water's movement, that the wind over the surface has, is complex to model accurately due to its variation in z. Etling *et al.* (1985) consider a mixing-length approach to model the surface wind stress, with results similar to the previously published analytical solutions. On the other hand, Baines (1974) uses a constant expression for the wind to show that the wave-induced drag coefficient reaches a maximum when wind speed is $2\sqrt{gh}$. In this thesis, as was done by Jha (2006), we make a simplifying assumption that forcing due to wind can be approximated by the expression

$$\tau = (\tau_x,\, \tau_y) = (c_w w^2 \cos\alpha,\, c_w w^2 \sin\alpha),$$

where c_w is the wind friction coefficient (assumed constant), w is the wind speed, and α is the angle of the direction of the wind with respect to the x-axis. The size of c_w is taken to be $\mathcal{O}(10^{-3})$ (as has been done by Trenberth *et al.* (1989), Yu and O'Brien (1991), and Butman (1978)), and the wind speed w may typically vary between 1 and $10\,\mathrm{ms}^{-1}$. Mete Uz *et al.* (2002) note that when wind speed is greater than $10\,\mathrm{ms}^{-1}$, larger waves are generated, which increases the drag coefficient, hence the assumption of a constant wind friction becomes invalid. As typical values in Moreton Bay are generally less than $10\,\mathrm{ms}^{-1}$, a constant drag coefficient is a valid approximation.

2.1.2 Shallow water equations in a circular domain

A special case where analytical solutions to the SWEs are available is when the domain is axisymmetric and the flow is steady, and therefore independent of time. Solutions to some other special cases of shallow water in circular coordinates are studied in the following papers.

Numerical solutions to several problems in a dish-shaped basin of 192 m diameter and 1 m maximum depth are given by Borthwick *et al.* (2001a) and Borthwick *et al.* (2001b). In one of these tests, wind of $10\,\mathrm{ms}^{-1}$ is simulated until steady-state is achieved. Their calculations are made on quadtree grids (square elements), and the solution shows two areas of circulation, which is in good agreement with the analytical solution for the same problem studied by Kranenburg (1992). This test is reproduced in this thesis in Chapter 5, using unstructured grids (triangular elements), but results are very similar.

Slim and Huppert (2004) use the phase plane method to investigate the existence of similarity solutions to the shallow water equations in two concentric cylinders filled with stationary fluids of different densities. As the inner cylinder is removed, gravity currents propagate inwards. Unique solutions are found for non-negative, finite Froude numbers under the assumption of regularity of height and velocity.

A cylindrical hydraulic jump occurs when a jet of water is directed vertically downwards onto a flat plane and creates a circular hydraulic jump. Bohr *et al.* (1993) show that the assumption of a viscous force in the fluid is necessary to describe the hydraulic jump, and derive a scaling relation for the radius of the hydraulic jump. Ray and Bhattacharjee (2005) use the same equations as Bohr *et al.* (1993) (*i.e.* SWEs in cylindrical coordinates with viscosity) and carry out a linear stability analysis on the time-dependent equations and give a condition on the Froude number for the jump to be able to form.

2.2 Numerical schemes

Numerical methods for PDEs has an extensive literature, helped by the relatively recent development of computers. Some of the earliest contributions come from Courant *et al.* (1928), von Neumann and Richtmeyer (1950), Courant, Isaacson and Reeves (1952) and Godunov (1959), although finding solutions via discretisation goes back to Euler (1707-1783) and his method for solving ODEs (Ordinary Differential Equations).

There is a choice of methods for approximating the solutions to partial differential equations; three very well-established methods are now described:

Finite element method (FEM) The advantage of this method is that it can easily be applied to domains with irregular boundaries.

The FEM is usually used for structural and mechanical problems because such problems can be approximated quite well with fewer cells. On the other hand, fluid problems tend to need much higher numbers of cells to obtain a good approximation, and so other methods are preferable.

Finite difference method (FDM) This method is widely used because of its simplicity in implementing. Shocks can be captured effectively via a flux limiter. However, it is unsuitable for problems with irregular boundaries.

The FEM and FDM differ in that the first makes an approximation to the *solution*, and the second is an approximation to the *differential equation*.

Finite volume method (FVM) This method has become popular as it has the simplicity of the FDM in programming, it is more efficient than the FEM, and is better-suited to 2D problems. Since it is naturally mass-conserving, it is particularly favoured in hydrodynamic modelling for systems of conservation laws. As it does not require a coordinate transformation when applied to unstructured meshes, it is useful for modelling natural domains with irregular boundaries.

The FVM is similar to FDM in that it makes calculations based on values at a point, but different in that the value used is the average value over the element (volume). The FVM uses the integral form of the equation.

A fuller description of all three methods can be found in Hirsch (1984), and very detailed information on implementing FVM across a range of problems can be found in LeVeque (2003).

Two further methods are sometimes used for solving PDEs. One is the spectral method, which approximates the solution on the whole domain using a high-order Chebyshev polynomial (in contrast, the finite element method approximates the solution as a series of overlapping low-order polynomials). It performs best when the geometry of the problem is smooth (Boyd (2000)), and is therefore unsuitable for shock-capturing methods. The other is the method of lines, or the method of characteristics, where the system is discretised in all but one variable,

reducing it to a system of ordinary differential equations and solutions are found along these lines numerically by using an explicit or implicit solver. See for example Ockendon *et al.* (1999).

Riemann2D was developed for short time problems, so it was important to use a shock-capturing scheme. It was intended for use on real domains and was required to fit easily to irregular boundaries. Hence the FVM was the most appropriate.

2.2.1 Properties of hyperbolic problems

PDEs can be classified according to their eigenvalues as either hyperbolic, parabolic, or elliptic. The SWEs have real and distinct eigenvalues (for $h > 0$), meaning the system is hyperbolic.

Hyperbolic systems have two significant properties that a successful numerical scheme should reproduce: one is that they have an inherent direction of signal propagation; the other is that they can admit discontinuous solutions. The propagation of information can be represented by characteristic curves (or just characteristics). Where these characteristics intersect corresponds to a shock in the system. A shock can be due to a discontinuity in the initial data (the Riemann problem, see below), or it can develop even when the initial data is smooth.

2.2.2 The Riemann problem

The Riemann problem is a hyperbolic PDE together with piecewise constant initial data having a single discontinuity, which is written mathematically as

$$\mathbf{q}_t + f(\mathbf{q})_x = 0$$
$$\mathbf{q}(0,x) = \begin{cases} q_L & \text{if } x > 0 \\ q_R & \text{if } x < 0, \end{cases}$$

for the vector of state variables $\mathbf{q}$, and a flux function f. It is possible to devise a numerical scheme by imposing a grid on the domain of calculation and assuming the state value in each cell or element in the grid is constant, so that wherever two cells meet there is a local Riemann problem. Such a scheme is called a Godunov-type Riemann solver. A Godunov scheme solves a Riemann problem exactly (Glaister (1990)). Each pair of cells is defined as being to the left (subscript L) or the right (subscript R) of the discontinuity. The state values of these cells are known as the left state (q_L) and right state (q_R).

This is the problem from which *Riemann2D* gets its name. It treats a problem by considering pairwise elements and solving across the interface.

2.2.3 Numerical schemes for hyperbolic problems

One of the most influential researchers in the field of numerical methods is Godunov. His famous theorem of 1959 states that a linear monotone scheme is at most first-order accurate. A monotone scheme is either only increasing or only decreasing and thus it does not generate any new maximum points or minimum points (extrema). It does not permit spurious oscillations.

Some of the most popular schemes developed in the 1970s and 1980s are described by Hirsch (1984) in the following way. (Most references of the next three subsections can be found in Hirsch's book, p128.)

Centred schemes

Space-centred schemes were among the earlier developments in the context of numerical methods. These include the first-order accurate Lax-Freidrichs scheme (1954), the second-order Lax-Wendroff (1960) and the two-step methods of Richtmeyer and Morton (1967) and MacCormack (1996). The Lax-Wendroff method is known to suffer badly from spurious oscillations.

Upwind schemes

Upwind schemes for the shallow water equations have been used with good results by Bermúdez and Vázquez (1994), Bermúdez *et al.* (1998), Yoon and Kang (2004), and Vázquez-Cendón (1999), among others. In fact, an upwind scheme is preferable to a centred one in order to ensure stability and the correct propagation velocity (Bermúdez and Vázquez (1994), Vázquez-Cendón (1999)).

Riemann2D uses an upwind method that is part of the Godunov class of solvers. Godunov's original solver of 1959 is a first-order scheme based on the exact solution to the Riemann problem. Van Leer (1979) increased the accuracy to second-order by adding a limiter. Approximate solvers (approximate solutions to the Riemann problem) have been proposed by Roe (1981), Engquist and Osher (1980) and Harten, Lax, Van Leer (1983). In all these methods a Riemann problem is solved, so they are sometimes known as Riemann solvers. In particular, the work of

Roe (1981), due to its stability and accuracy, is used in the design of the numerical scheme of this thesis, and is discussed further in the next section.

High-resolution schemes

High-resolution schemes aim to achieve second- or higher-order spatial accuracy in regions of smooth flows. They model areas of shocks or discontinuities better than a first-order scheme and significantly reduce spurious oscillations. Central schemes were made high-resolution in the eighties by Davis (1984), Roe (1985) and Yee (1985,1987). High-resolution schemes that are upwinded include those of Van Leer (1974), Van Leer (1979), Harten (1983), (1984) and Osher (1984).

Since, according to Godunov's theorem, monotonicity is required in order to ensure that no new extrema are generated in a linear scheme, a *nonlinear* term can be introduced to make the scheme second-order accurate. von Neumann and Richtmeyer (1950) proposed adding a numerical 'anti-diffusive flux' or 'artificial viscosity' in order to prevent the spurious oscillations that could occur [1]. Adding a term that damps oscillations does not mean the scheme can be said to be high-resolution, as it does not prevent the oscillations from being generated (Hirsch (1984), p127).

Boris and Book (1973) were the first to add a term that 'limits' the flux [2]. Depending on the gradient of the solution in a cell, the limiter acts more (on a steep gradient) or less (on a shallower one) meaning that oscillations are prevented around very high gradients, while still maintaining accuracy in regions of small slope. Many limiters have since been developed. These include the superbee limiter due to Roe (Sweby (1984)), Van Leer's limiter, Chakravarthy and Osher's limiter, and the MLG (maximum limited gradient) limiter due to Batten *et al.* (1996) (which has been shown to reduce to the superbee limiter in 1D, see Jha (2006) and references therein). A detailed description and comparison of some of the most popular limiters is given by Sweby (1984), which is also summarised in Section 4.1.4. See also Jha (2006). For most of the

1. These methods were developed for finite difference schemes.
2. The work of Boris and Book (1973) was carried out for finite difference methods. Their algorithm "employs no adjustable artificial viscosity of the von Neumann type", but instead acts in an antidiffusive manner to correct any undershoots or overshoots.

numerical tests in this thesis, the MLG limiter is used as it is the most effective at preventing numerical smearing.

Godunov's theorem on monotonicity was generalised in a paper by Harten (1983), in which a general condition was given for a nonlinear scheme to not generate unphysical oscillations as being total variation nonincreasing (TVNI). A scheme is said to be TVNI if is satisfies

$$TV(q^{n+1}) \leqslant TV(q^n),$$
$$\text{where} \qquad TV(q) = \sum_{j-\infty}^{\infty} |q_{j+1} - q_j|,$$

where the superscript n denotes the nth time step and the subscript j denotes the jth cell (space). The more commonly-used form is total variation diminishing (TVD) (where the $\leqslant$ is replaced with $<$). The condition guarantees a maximum principle on the discrete solution and that mass and other physically relevant data may not become unbounded (Batten *et al.* (1996)). The exact solution to a hyperbolic problem is TVD and thus any numerical scheme that hopes to approximate the solution should also have this property. A TVD scheme must also satisfy the entropy condition if it is to select a physical solution (Yee *et al.* (1985)). The condition on entropy is that it must always remain constant or increase, never decrease, and that it is not conserved over shocks.

2.2.4 Roe's scheme

Roe (1981) was one of the first to propose solving an approximate problem exactly instead of the exact problem approximately. Because of the success of this approach it has become one of the most popular methods to use, especially in problems involving complicated geometry (Namin *et al.* (2004)), and is thus an obvious choice for the *Riemann2D* code. Roe's approximation must satisfy the U-property (uniform property), defined by Roe as the following. For a system $\mathbf{q}_t + \mathbf{F}_x = 0$, which can be written as $\mathbf{q}_t + A\mathbf{q}_x = 0$ for $A = \partial\mathbf{F}/\partial\mathbf{q}$, we define an approximation matrix $\tilde{A}$ that satisfies

i) $\tilde{A}$ is a linear mapping from $\mathbf{q}$ to $\mathbf{F}$,

ii) $\tilde{A}(\mathbf{q}_L, \mathbf{q}_R) \to \tilde{A}(\mathbf{q})$ as $\mathbf{q}_L \to \mathbf{q}_R \to \mathbf{q}$,

iii) $\tilde{A}(\mathbf{q}_L, \mathbf{q}_R) \cdot (\mathbf{q}_L - \mathbf{q}_R) = \mathbf{F}_L - \mathbf{F}_R$ for all $\mathbf{q}_L$, $\mathbf{q}_R$,

iv) the eigenvalues of $\tilde{A}$ are all linearly independent,

where subscripts L and R denote left and right states, respectively. It is a relatively easy task to find an $\tilde{A}$ that satisfies properties i), ii) and iv), but to satisfy iii) is most difficult. However, for the 1D SWEs, $\tilde{A}$ can be found (*e.g.* Glaister (1988a)) by replacing the state variables with the so-called Roe averages, given as

$$\begin{aligned}\sqrt{\tilde{h}} &= \sqrt{h_L} + \sqrt{h_R} \\ \tilde{u} &= \frac{(hu)_L/\sqrt{h_L} + (hu)_R/\sqrt{h_R}}{\sqrt{\tilde{h}}}.\end{aligned}$$

These have also been given for the 2D shallow water equations in, for example Namin *et al.* (2004) and Bradford and Sanders (2002).

2.2.5 Splitting

When formulating a numerical scheme, one approach is to perform a 'splitting' of the flux term, whereby the flux term is divided in two, dealt with numerically, then the result recomposed at the end of the calculation.

For a system of conservation laws such as

$$\mathbf{q}_t + f(\mathbf{q})_x = 0,$$

where $\mathbf{q}$ is the vector of state variables, and $f(\mathbf{q})_x$ is the flux function, discretisation allows us to write

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2}^n - F_{i-1/2}^n\right),$$

for some numerical flux F and where Q_i^{n+1} is the solution calculated at the next timestep at a point in space i and at time $n+1$. The flux F is defined as a function $\mathcal{F}$ of left (Q_i) and right (Q_{i+1}) states. We could, for example, define $\mathcal{F}$ via

$$Q_{i-1/2}^n = \mathcal{F}(Q_{i-1}^n, Q_{i+1}^n) = \frac{1}{2}\left(f(Q_{i-1}^n) + f(Q_i^n)\right).$$

But it turns out that this is unstable in general. A more stable scheme due to Lax-Friedrichs is obtained by performing the splitting as

$$\mathcal{F}(Q_{i-1}^n, Q_{i+1}^n) = \frac{1}{2}\left(f(Q_{i-1}^n) + f(Q_i^n)\right) - \frac{\Delta x}{2\Delta t}(Q_i^n - Q_{i-1}^n).$$

Or alternatively due to Richtmeyer

$$F_{i-1/2}^{n+1/2} = f(Q_{i-1/2}^{n+1/2})$$
$$\text{where} \qquad Q_{i-1/2}^{n+1/2} = \frac{1}{2}\left(Q_{i-1}^{n} + Q_{i}^{n}\right) - \frac{\Delta t}{2\Delta x}\left(f(Q_{i}^{n}) - f(Q_{i-1}^{n})\right).$$

There are many ways of splitting the numerical function and the best choice depends on the problem to be tested.

2.2.6 Structured and unstructured meshes

The irregularity of the shape of the boundary of a natural domain means that the choice of mesh becomes an important consideration. A structured quadrilateral mesh (*i.e.* based on a Cartesian grid) applied to such a domain may lead to difficulties in fitting the elements at the boundaries, and is thus not always chosen for natural domains. Nevertheless, good results have been found with quadrilateral meshes by working on either, or both, of two principles:

a) that the mesh is fine at the boundary

b) that the cells are aligned to fit the boundary, for example in a problem with a circular domain, cells are aligned radially.

The 2D shallow water equations are solved on a bifurcated channel with a quadrilateral mesh (based on principle (b)) by Wang *et al.* (2003), and good agreement is reported with experimental data. Borthwick *et al.* (2001a) and Borthwick *et al.* (2001b) use quadtree meshes based on principle (a) in a circular basin, and find good agreement with previously published results. Feng *et al.* (2006) and Lin *et al.* (2003) solve the 2D circular dam break problem. Lin *et al.* (2003) also use cells that are radially aligned, while Feng *et al.* (2006) use a Cartesian grid. Results are shown to be stable in both cases.

Whereas structured meshes are based on the intersection of parallel lines, an unstructured mesh is not. In this thesis we use unstructured triangular meshes, as they are well-suited to problems where flexibility in fitting to irregular domains is desirable. Namin *et al.* (2004) use both "cell-centred" and "mesh vertex" approaches for unstructured triangular meshes and find reliable results in both cases, the difference being that the cell-centred mesh takes $2\sqrt{2}$ times longer than the vertex mesh as it has more computational points. Brufau and Garcia-Navarro

(2000) and Yoon and Kang (2004) use unstructured meshes and an FVM to simulate a dam break. The former paper applies the model to a channel with a constriction, and the numerical solutions show good agreement with experimental results. It is found that some more work is required on the choice of boundary conditions for this model, although a very fine mesh is found to reduce these problems. The latter shows how the connectivity – the number of cells meeting at one node – of the mesh can affect the stability of the solution. Bermúdez *et al.* (1998) also use an unstructured mesh, which they test on a partial dam break problem and find to be reliable. Anastasiou and Chan (1997) use Roe's solver on unstructured meshes for a partial dam break, a circular dam break, flow over a step, and jet-forced flow in a circular reservoir, and very good agreement is found with previously published results.

2.3 Shallow water applications

Numerical schemes have been used to model the shallow water equations with applications to bays, lakes and harbours. Below is a selection of papers describing such schemes. The schemes cited are based on finite volume methods applied to unstructured triangular meshes, unless stated otherwise.

The Bay of Fundy near New Brunswick/Nova Scotia/Maine, North America, has been the subject of several studies. It is of particular interest because of its extreme tidal range – about 17 metres on average and arguably the largest in the world. The area of the Bay is 16 000 km^2 and is generally less than 200 m deep at any point. Heaps and Greenberg (1974) use a finite difference scheme to solve the shallow water equations and to describe the M_2 tidal behaviour [1] in the Bay. Amplitudes and phases at the boundary are determined by observed field data and the simulation is made over nine tidal cycles. The study was motivated by the proposal to build barriers for electric power generators in order to calculate the tidal power and to optimise the barrier schemes. The authors also provide a simplified analytical model based on the SWEs that agrees with the numerical results.

Heaps and Jones (1976) were also concerned with the construction of cross-channel barriers in the Bay of Fundy, and used results from a numerical tidal model to predict the phase and

1. The M_2 notation is used to describe the (dominant) effects of the Moon on the tide. It is a practical way to simulate the complex force of the tide using a simple sine function.

amplitude of the M_2 tides. The poor comparison with observed M_2 data, the authors say, is due to the coarseness of the grid used in their model.

Rasmussen and Badr (1979) were some of the first to validate the results from a numerical model for which measured field data were used as boundary conditions. They use a mathematical model developed by the Water Resource Branch, Ontario Ministry of the Environment, applied to Hamilton Harbour, Ontario, which is approximately 7 km long and 3 km wide (it is not stated what type of model this is). The major features of the flow in Hamilton Harbour are found to be reasonably well modelled, while the more detailed features are not as well reproduced. The authors suggested that it is likely that with a smaller time step and a finer mesh a better description of these detailed features would be produced, but since computer power was limited at that time, this would have been at an unacceptable computing cost.

Bermúdez *et al.* (1998) develop a first-order accurate upwind scheme with flux splitting and test it on a partial dam break and the authors report good agreement with published results. The model is then applied to tidal flows in the Pontevedra ria, Galicia, Spain (15 km long, 5-10 m deep) with slip boundary conditions on the coastal boundary and specified h at the ocean boundary. The numerical results show that velocities in the ria resulting from the tidal flow are between $0.004 - 0.15\,\mathrm{ms}^{-1}$.

Zhou *et al.* (2001) use a Surface Gradient Method (SGM) which is comprised of a Godunov-type solver with a predictor-corrector method and a quadrilateral mesh. Results for a tidal wave over an irregular bed, steady flow over a bump, quasi-steady flow (all 1D problems, 1.5 km, 24 m and 1 m long, respectively) and flow through a constricted channel (2D problem) are compared with analytical solutions and excellent agreement is found.

Lin *et al.* (2003) modify the well-known MacCormack scheme with flux-splitting to estimate the numerical flux. The splitting means the eigenvalues of the Jacobian matrix are not required hence programming the scheme is simpler. Four types of splitting are tested on four tests: a 1D dam break, an oblique hydraulic jump, a partial dam break and a circular dam break. All schemes perform well, but the 'Liou-Steffen' splitting performs best, and is chosen to reproduce a lab experiment of a dam-break with a sloped, partially dry bed, for which good agreement is reported.

Gómez-Valdés *et al.* (2003) use a finite difference model to investigate the residual tidal currents in the Ensenada de la Paz, Mexico, a bay of about 8 km square. Their results are calibrated with observed surface elevation and currents from that bay, and good agreement is found.

Wang *et al.* (2003) apply their finite-volume TVD scheme, using structured quadrilateral meshes, to a partial dam break, to a circular dam break with flat bed and no bed friction, and finally to a dam break in two channels – one trifurcated, one bifurcated. For the first two tests, good agreement is reported with other published results. In the case of the channels, general flow patterns are considered reasonable but no laboratory data were available for specific comparisons to be made, so the authors suggest further study is required.

Yoon and Kang (2004) test their Harten-Lax-vanLeer approximate Riemann solver on an oblique hydraulic jump problem with different limiters and types of grid (well-connected and distorted) and show their choice of limiter exhibits smaller oscillations than the superbee and minmod limiters even on a distorted grid. A dry-bed dam break problem is tested and results compared to published experimental results, showing reasonable agreement in flow depth and arrival time of the wavefront. They then apply the model to the Malpasset dam in France, which failed in 1959, meaning that data are available for surface elevation and velocity at various points down the valley. Predicted results are very close to the measured data of 1959.

Feng *et al.* (2006) combine the fourth-order central weighted essentially non-oscillatory (CWENO) reconstructions with a central-upwind scheme. They use structured meshes to test their scheme on 1D and 2D problems with varying bottom topography and a 2D circular dam break problem on a domain of 20 m square. Comparison with exact results shows that these methods have high resolution, non-oscillatory behaviour and robustness.

Castro *et al.* (2006) present a one- and two-layer shallow water model, and use it, with an unstructured triangular mesh, to compare with analytical solutions and laboratory experiments. Then the model is used to simulate a lock-exchange experiment in the 5 km long Strait of Gibraltar. Numerical results are in good agreement with the hypothetical steady state proposed by Farmer and Armi (1988). The same authors, in a subsequent paper (Castro *et al.* (2007)), apply a two-layer shallow water model to the Strait of Gibraltar to simulate the tide at the two open ends based on observed tidal data. Once again, results show good agreement with Farmer

and Armi (1988), and also with experimental measurements for the steady-state simulation. For the tide simulation, agreement with measurements is found to be quite good; in particular the travelling bores are found to be well simulated.

Wetting and drying

The following papers are all concerned with modelling the phenomenon of wetting and drying, which is a particular consideration when modelling tidal flows.

Bradford and Sanders (2002) develop a model using a predictor-corrector technique, with Roe's scheme, to evaluate the flux. It deals with wetting and drying of cells by setting velocities in a cell to zero if the depth of water in that cell becomes smaller than a specified tolerance (in frictionless problems, the tolerance is 1×10^{-4} m). The model is tested on a frictionless dam break, 120 m long, with a sloping bed and dry bed ahead of the dam, and on an upcoming wave around a conical island on a 24×24 m domain. The results compare well with experimental results and previously published results.

Namin *et al.* (2004) use the results of experiments on a 20 km-long 2D dam break with varying bed slopes and alternately wet and dry beds downstream of the dam. Detailed comparisons are made between measured (experimental) and predicted (numerical) values, and reliable predictions from their model based on Roe's Riemann solver are reported. The model is applied, with some modification to the flux in areas of rapidly-varying topography, to the Ribble Estuary, northwest England, which is about 15 km long. The boundary conditions are set using data from the Proudman Oceanographic Laboratory's Irish Sea model. Due to the large tidal range in this estuary, there are large areas subject to wetting and drying, which the model copes well with, and typical flow velocities are found to be of $0 - 5\,\mathrm{ms}^{-1}$ with a maximum of $12\,\mathrm{ms}^{-1}$. This is in contrast to the magnitudes found by Bermúdez *et al.* (1998) ($0.004 - 0.15\,\mathrm{ms}^{-1}$), and can be explained by the fact that water flows much faster over shallow bathymetry.

Greenberg *et al.* (2005) use a three-dimensional wave model and a finite element scheme to simulate flooding and drying of cells by reducing the proportion of the column that is 'active' when water depth is below 10 m (water depth varies from 0 to 45 m, length 51 km square). The model is applied to the Quoddy region near the Bay of Fundy, New Brunswick and incorporates

an approximate tidal force. It is found to have good results when compared with available observational data, and the results demonstrate good quantitative behaviour.

Liang *et al.* (2006) present three improvements to an existing numerical code, which tests show produce either the same or better results than the original. Their new finite difference model is used on a quadrilateral mesh to simulate the tidal flow in Poole harbour, a natural coastal basin about 10 km long, in Dorset, UK. A wind of $3\,\mathrm{ms}^{-1}$ is blown across the surface and the tidal data fed in are based on measurements at the harbour mouth. Close agreement is found with the water level, and although the velocity is less exact, it still gives reasonable agreement. This is attributed to the highly irregular topography in the harbour and local conditions such as wind speed, bed roughness and the accuracy of the field measuring equipment.

Liang and Borthwick (2009) use a quadtree mesh and a Godunov-type solver and test their scheme on 1D and 2D flow over a hump (domains of 25 m and 2×1 m), 2D flow over three humps (30×10 m), and sloshing in a vessel with parabolic bottom (5 km radius, 10 m deep). Results compare well with analytical solutions or previously published numerical predictions.

2.3.1 Pollutant transport

Pollutants may come from an agricultural source, *e.g.* fertilisers or pesticides, which run off the land via rainfall, and enter the water system by a river or stream. Alternatively, they may have their source in a factory or manufacturing plant, where chemicals used in processing are flushed into the rivers without first being treated. Effective management of these pollutants is essential to evaluating coastal action plans (Hodge *et al.* (2005)). Moreton Bay's ecosystem supports rare wildlife such as the giant turtle and the dugong, whose habitat of seagrass is already threatened with the degradation of water quality and urban development (Haynes (2001)).

The following papers demonstrate how effectively wind is at transporting contaminant, water particles, or even eroded bed particles.

Neill *et al.* (2008) use the Proudman Oceanographic Laboratory Coastal Ocean Modelling System and a longshore sediment transport model to provide insight into the mechanisms associated with the erosion of dunes and beach levels (bed evolution) of Tremadoc Bay in northwest Wales. The model incorporates wave, tidal, longshore transport, total transport and bed level change modules with the aim of determining how effectively it could reproduce the

bulk flow features that were observed. Results of the long-duration simulations including the effect of high-frequency wind effects agree well with the beach profile data over six-month time periods for the seven years of measurements between 1997 and 2005.

The erosion in Massachusetts Bay, in northeast USA, is due mainly to bottom stress due to surface waves, according to Butman *et al.* (2008). This is because the prevailing northeasterly wind creates long waves, which have a strong effect on the bottom. A metric is proposed to rank storms according to their duration and bottom stress magnitude, using observed data from 1990-2006 and it is found that over the 17-year study period, Massachusetts Bay suffered 151 storms, and the three most severe of these were so-called northeasters.

Liang *et al.* (2009) simulate the effect of wind blowing across the surface of a 2 km long lake, located in northeastern New Brunswick, Canada. The flow field resulting from the numerical model is decomposed and stored using Singular Value Decomposition (SVD) to reduce the quantity of data stored. Then particle trajectories are obtained by numerical integration of the advection equations using a Runge-Kutta algorithm with the velocity field reconstructed from a few dominant SVD modes. Under a northeasterly wind, one gyre is produced, while under a northwesterly, there are two. This is due to the geometry of the lake, which is longer in the NE-SW plane than it is wide. The particle paths are seen to be moved roughly in the paths of these gyres over a simulation time of around 10 hours. The authors judge that the model provides a potentially useful tool for investigating transport processes in shallow ecosystems, such as algal blooms in shallow lakes or reservoirs.

In the model used in this thesis, we assume that the presence of dissolved pollutants does not significantly affect the flow of the water, *i.e.* that they are a passive element in the system. In this simple model, we add the following advective equation to the usual shallow water equations to model concentration

$$\frac{\partial(hc_i)}{\partial t} + \frac{\partial(huc_i)}{\partial x} + \frac{\partial(hvc_i)}{\partial y} = 0,$$

where c_i denotes the ith class of concentration, which could be either pesticide or fertiliser concentration or even suspended sediment. The above equation models the conservative transport of contaminants only – chemical reactions are all ignored. With this model, pollutants entering a bay may be traced to see how long they stay in the bay and what path they take.

Lin and Falconer (1997) and Benkhaldoun *et al.* (2007) use the advection-diffusion equation to model contaminant transport by modifying the right hand side to include both turbulent dispersion effects and the presence of contaminant sources or sinks as

$$(hc)_t + (huc)_x + (hvc)_y = \nabla \cdot (h\mathbf{D}\nabla c) + hQ, \tag{2.5}$$

where c is the concentration, $\mathbf{D}$ is an empirical diffusion matrix, given by Lin and Falconer (1997) as

$$D_{xx} = \frac{(k_l u^2 + k_t v^2)h\sqrt{g}}{C\sqrt{u^2+v^2}} + D_w, \quad D_{yy} = \frac{(k_t u^2 + k_l v^2)h\sqrt{g}}{C\sqrt{u^2+v^2}} + D_w$$
$$D_{xy} = D_{yx} = \frac{(k_l - k_t)huv\sqrt{g}}{C\sqrt{u^2+v^2}},$$

where k_l is the longitudinal dispersion coefficient, k_t is the turbulent diffusion coefficient, C is the Chézy friction number, D_w is the wind-induced dispersion coefficient, and Q is the pollutant source or sink, which is taken to be zero.

Lin and Falconer (1997) use the ULTIMATE QUICKEST scheme with rectangular, body-fitted grids on which to solve the augmented system, and the model is applied to the Humber Estuary, UK, to trace a pollutant released at low tide. The model is verified for two test cases for which analytical solutions exist, and it is found that both versions of the model give stable results that compare well to the analytical solutions. Benkhaldoun *et al.* (2007) use a non-homogeneous Riemann solver with the advection-diffusion equation (2.5) above, with a predictor-corrector method on unstructured adaptive triangular grids, and with boundary conditions enforced by setting values at the boundary, finding reliable and realistic results for two-dimensional dam break flow over a step in a 12×1 m domain. A test for contaminant is carried out on a 9×9 km mesh and the concentration of contaminant is found to be very well preserved, also comparing well with other published results. The tests are repeated with the minmod and the VanAlbada limiters, and it is shown that the results with the minmod limiter exhibit more numerical diffusion. Both Benkhaldoun *et al.* (2007) and Periáñez (2004) apply their shallow water model to pollutant transport in the Strait of Gibraltar. In particular, Periáñez (2004) solves the usual 2D shallow water equations with an explicit finite difference scheme, then includes the effects of the tide after a stable solution is achieved. The streamlines are plotted, and it is seen that a westerly wind moves a pollutant released out of the Strait

faster than an easterly, because the westerly winds are on average stronger than the easterly. This agrees in principle with Butman *et al.* (2008), Neill *et al.* (2008) and You (2005b), who find from measurements made in Massachusetts Bay, Tremadoc Bay and Moreton Bay, respectively, that higher wind speed corresponds to higher levels of transport.

Kachiashvili *et al.* (2007) use a finite difference approximation to solve a one-, two- and three-dimensional convection-diffusion equation, where the polluting substances are nitrite and phosphite, and apply the model to two river systems in Georgia that flow into the Black Sea, estimating the effect of agricultural practices on the pollution in the rivers, and finding good agreement with measured values of pollutants in the rivers.

2.4 Brisbane and Moreton Bay

Brisbane is the state capital of Queensland and is located in the southeast region of the state (see Figure 2.2). It has a population of just over a million people that grew at 1.6% in the period September 2006 - September 2007, a rate just above the national Australian population growth of 1.5%, according to the Australian Bureau of Statistics. More than a million people are expected to move to the Brisbane area over the next 25 years (Olley *et al.* (2006)). The climate there is humid and subtropical, with an average temperature of 20°C in June and 29°C in January, and a maximum of 43.2°C (recorded in January 1940 [1]) and minimum of −0.1°C in July 2007 (Bureau of Meteorology).

Water is a major issue in Australia, and in this section we consider some of the most important aspects to the supply and maintenance of the water system.

2.4.1 Erosion

Water quality is directly influenced by erosion, not only from the point of view of particles themselves that enter the water system, but that pollutants bind to eroded particles and are therefore more likely to remain a threat to the surrounding environment (Dunn *et al.* (2007)). Erosion is a significant problem in Australia and so although this thesis is more concerned with

1. www.nationmaster.com/encyclopedia/Brisbane

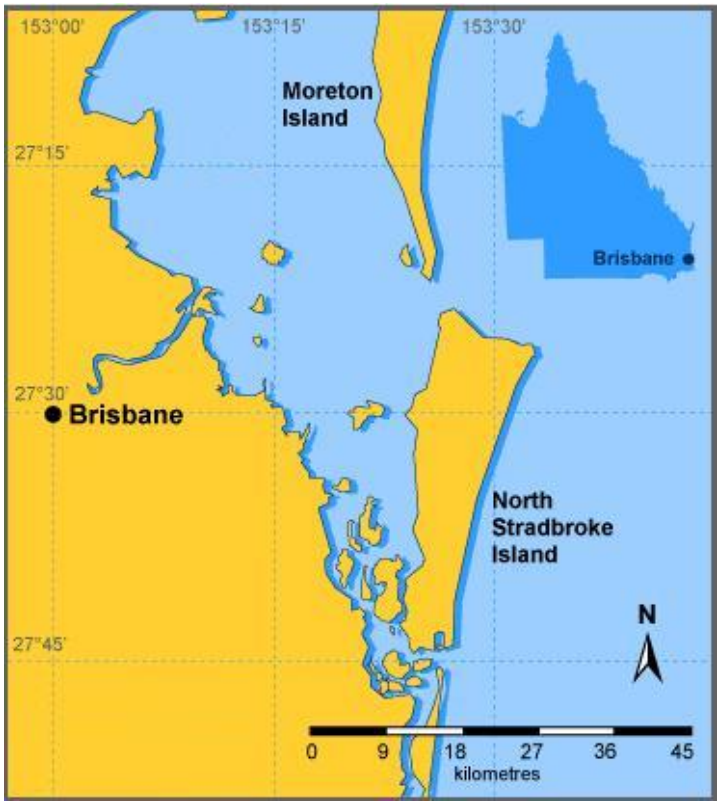


Figure 2.2: Map of Moreton Bay and Queensland. Courtesy of `epa.qld.gov.au`. The city of Brisbane is located on the southeast coast and Moreton Bay is to the east of the city.

pollutant transport than erosion, we take a detour here into the issues surrounding increased levels of sediment.

Extensive removal of trees (only 26% of the original vegetation remains in the catchment area of Moreton Bay, Wallbrink (2004), Bolton (1981)) since Western colonisation in the 1800s has led to the degradation of the soil, as tree-roots that previously acted to hold soil together have been now removed. Gully erosion is now extensive and is especially significant on granitic soils (Wallbrink (2004)).

2.4.2 Drought

Average rainfall in Australia is 420 mm per year,[1] although this varies greatly from year to year and from region to region (Chanson (1998)).

In periods of drought,[2] agricultural industry suffers from reduced crop output and cattle and sheep fail to thrive, which can lead to a rise in inflation and resulting economic concerns (Heathcote (1974)). Restrictions are placed on industries and individuals as to water use, and dry land combined with heavy farming or high winds can lead to dust storms.

Figure 2.3 comes from the Australian Government's Bureau of Meteorology website and shows the areas of the continent that have been affected by drought since 1864. As the figure below the maps shows, severe droughts have been experienced as frequently as every four years or as far apart as 38. Some last for several years, others only a few months. Every region in Australia is at risk from drought.

Drought can lead to a decrease in water quality as a result of reduced dilution of contaminants entering the water system.

2.4.3 Moreton Bay

Moreton Bay itself lies to the east of Brisbane and is approximately 90 km long and 30 km wide, with a planform area of about 1 200 km^2. It is between on average 6 and 10 m deep, increasing to around 30 m at the ocean entrance. It is enclosed on the east by two large, sandy islands: Moreton Island and North Stradbroke Island. The Bay is open to the ocean at the northern section (an inlet of about 15 km across), and between the two Islands (a smaller gap known as the Southern Passage of about 1.6 km You (2005b)). Five main rivers feed it, the largest of which is the Brisbane river, and there are many minor rivers besides the five main ones (Wallbrink (2004)).

1. For comparison, rainfall in the UK averages around 1 000 mm per year, comprising 800 mm for England, 1 300 mm for Wales, 1 400 mm for Scotland and 1 000 mm for Northern Ireland. Source: Met Office; Centre for Ecology and Hydrology, Wallingford.

2. Drought is defined by the Bureau of Meteorology by considering the rainfall of the past three months and comparing with the average rainfall rates. If the figure lies in the 10th percentile, it is considered a drought period. However, rainfall is only one measure of drought, and the responsibility of the State Government to declare a drought must take account of factors other than rainfall.

Management schemes to obtain, preserve, and re-use water are being planned. The Queensland Water Commission [1] have investigated possible sites for desalination plants in Southeast Queensland. This follows the desalination plant that opened in Perth in 2006, and which provides Perth with 17% of its drinking water requirements [2].

One of the most important studies currently being done on Moreton Bay is the RWQM1 (Receiving Water Quality Model) due to Shanahan (2001). It proposes a model that can be applied to any bay, although it is data from Morton Bay that are being used. The model considers chemical changes in the water to determine its quality. Bacterial decomposition of organic carbon requires oxygen, which impacts directly on the water, and the effect of factors such as settling rates of particles, the decay of dissolved matter, presence of phosphorous and algae are all considered. The study addresses the implementation of applications such as stormwater overflows, non-point-source and pathological pollution events, and catchment planning, coupling the hydrodynamic properties of the water with the quality of the water. It employs a two-dimensional finite element method that is still under development and improvement.

The above study forms part of a guide published by Moreton Bay Waterways and Catchments Partnership with the support of the Australian government (Howes (2006)). The guide is intended to be used as a Decision Support System (DSS) and explains the components of the system, which also includes an Environmental Management Support System and the Southeast Queensland Regional Water Quality Management Strategy.

There have been few attempts to apply a mathematical model of pollutant transport processes to Moreton Bay. Nevertheless the Bay is a popular study area, and some of the studies that have provided interesting results are summarised below.

Wallbrink (2004) measures fine sediment ($< 10\,\mu$m diameter) in the Brisbane and Logan rivers and finds that the sediment is likely to remain for 0-9 years and 0-21 years, respectively, with a mean of 5 years. The sediment is mainly derived from eroded subsoil and cultivated lands, so the author suggests that conservation works should initially focus on stabilisation of gully and channel networks.

1. www.qwc.qld.gov.au/Desalination+site+studies
2. The Water Corporation at www.watercorporation.com.au/D/desalination.cfm

Measuring the bed roughness of a sea bed is often difficult and inaccurate. Usually, the bed roughness is estimated by fitting measured current profiles to a logarithmic distribution. You (2005a) proposes a new way to estimate seabed roughness in Moreton Bay. Data from three study sites are used in a simple method based on mean flow velocities measured at two levels near the sea bed. It is found that in the two sites where tidal currents were strong, the seabed roughness could be accurately estimated, but estimations are less accurate where tidal currents are weaker.

You (2005b) conducts a study on fine sediment resuspension in Moreton Bay and finds that the main cause of resuspension is storm wind-waves, and not the tidal current or penetrated ocean swell, because of the relative shallowness of the water at the study site. Measurements of water depth, current speed and sediment resuspension are taken at one site over a period of three weeks and from these measurements an estimation of critical wave bed shear stress for sediment resuspension is given.

Hodge *et al.* (2005) use a portable underway water quality monitoring system to measure salinity and turbidity in the Brisbane and Daintree rivers during wet and dry periods (Daintree river is also on the Queensland coastline, north of Cairns). During both dry and wet periods, the Brisbane river shows a clear decrease in salinity as distance from the river mouth increases. At the same time, turbidity generally increases with distance, although in the dry period the increase is quite gradual and spikes in turbidity are small. In contrast, during the wet period the spikes of high levels of turbidity are high and apparently irregularly spaced. This turbidity is caused by particles being entrained by the rainfall that run over the land and into the river.

Dunn *et al.* (2007) measure distribution of nutrients (carried by the sediment) in Coombabah Lake, in the south of Moreton Bay, by sampling surface and subsurface sediments in November 2004 and again twelve months later to observe any changes that occurred in sediment concentrations. The effect of urban development in the surrounding area is not found to have an obvious impact on the accumulation of nutrients in the Lake, but the authors recommend that long-term sampling be carried out in order to better assess the importance of urbanisation to changes in the Lake.

An area in northern Moreton Bay known as Pumicestone Passage is the subject of study for Larsen (2007). Field measurements are taken for a number of points in this passage for surface

elevation and flow velocity that are then used to calibrate simulations made with the modelling software SMS and RMA2. Results are found to give better correlation with field data in the southern part of the passage than the northern, which is attributed to inadequate bathymetric information and the large amount of wetting and drying that occurs in this area. It is found that areas of higher bed friction reduce the tidal response, so that the tidal range is smaller than expected in these areas.

Field data, even if it is localised, is nevertheless valuable to a numerical model for the purposes of validation. The benefit of a mathematical model is to enable a greater understanding of the whole of the domain of study. It is also less costly than a field study, and not subject to weather conditions, quality of equipment, or human error. But it would be meaningless without field data to compare results.

2.5 Object Orientation and Java

Riemann2D has been written using an object-oriented (OO) approach. As such, it incorporates several important object-oriented principles, including *encapsulation*, *abstraction*, *polymorphism* and *inheritance* (see for example, Dietel (2005), Rajsbaum and Viso (2005) or Liu *et al.* (1996), among many others). These are not the only aspects that separate languages such as Java from procedural languages such as C and FORTRAN, but they are perhaps the ones that most distinguish the two branches.

OOP (object-oriented programming), as pointed out by Rajsbaum and Viso (2005) is a particular way of thinking, different from the procedural languages. It assigns properties and values to an object, and specifies ways in which objects may interact with each other. It is often said to try to more closely model the real world. For example, an object `myDog` may have properties (attributes) of `tail`, `fur`, `ears` and have methods (behaviours) `bark`, `fetch` and so on. It may interact with another object, `stick` via the method `fetch`. The idea of abstraction may be illustrated by noting that the object `myDog` is an instance of the conceptual class of `dog`. If my dog is a poodle, then `poodle` is a subclass of `dog` (`poodle` inherits all the attributes and behaviours of `dog`) and `myDog` is an instance of the `poodle` class.

Java is the OO language that *Riemann2D* has been written in. Released in 1995 by Sun Microsystems, Java has become the world's most widely-used object-oriented programming language (Dietel (2005)). One of Java's main advantages is its portability: it is portable to virtually all computing platforms because of the way it is compiled. This, and its efficiency and versatility, means it is the language of choice for many web applications and ideal for developing code as part of a team that is generic and easily extensible.

Because of its nature and the many internet applications that are written in Java, a large amount of information on OOP and Java can be found on websites such as

- `www.java.com`
- `www.java.sun.com`
- `ootips.org`
- `www.eclipse.org`

This last is the website for a Java IDE (Integrated Development Environment), Eclipse, that is used for all the simulations in this thesis. Eclipse is an open source environment that is freely downloadable and easy to use. More information is given in Appendix D.

2.5.1 Object Orientation for flows in bays

In the field of water quality modelling, object-oriented programming lends itself well to DSSs, since their aim is to model the interaction of complex actors such as tourism, agriculture, biological processes and so on. An actor in a system may be represented quite naturally as an object in a program. This project may be viewed as the mathematical class of a larger project for modelling the many processes that occur in a natural shallow water bay.

Many of the papers that use OO modelling applied to a bay are inter-disciplinary. OO lends itself very well to allowing discrete entities to interact effectively, which is necessary in a situation where there are many influences, both natural and man-made.

Spanou and Chen (2000) develop a model for the analysis of point-source pollution. The object-oriented approach means river networks can be represented, water quality assessed, low-flow curves estimated, river flows estimated, point-source effluents studied, and pollution control

schemes optimised. The model is validated with data from South Nation River catchment in Canada and the Upper Mersey River catchment in the UK, where good agreement is found.

Ludwig *et al.* (2003) emphasise the importance of an inter-disciplinary approach to water quality modelling. They establish a platform-independent structure of computational methods and interfaces to enable understanding between members of the project team, using UML (Unified Modelling Language) diagrams (see Dietel (2005)). The aim is the application to a catchment area of about $76\,000\,\mathrm{km}^2$ of the Danube river, Germany/Austria, but development of the model is still underway.

Newham *et al.* (2004) consider a dam and its catchment of an area of $985\,\mathrm{km}^2$ in New South Wales, Australia, and build a model that incorporates hydrologic, sediment, nutrient, management and economic submodels. The aim of the model is to enable the evaluation of different land management practices in terms of the nutrient and sediment delivery to the dam and the subsequent water quality. The model satisfies the specified criteria: simulation of hydrologic processes; identification of stream reaches; simulation of the impact of future management decisions; sensitivity to climate variation; and ability to incorporate socio-economic information on costs of remediation. These are deemed the minimum necessary to provide reasonable results, although the authors conclude more testing is needed to reach firmer conclusions.

Elshorbagy and Ormsbee (2006) also develop an OO model that allows different management scenarios to be simulated, as well as their respective effects on particle transport. The model is applied to a case study: a water catchment area of southeast Kentucky, USA, which is poor in data. As new data become available, therefore, OO allows for their easy incorporation into the code.

Becker and Jiang (2007) calculate flux-based contaminant transport in a GIS (geographic information system) environment by combining particle-tracking and first-passage-time modelling to yield a streamline-based interpretation of transport. The object-oriented approach is particularly suitable for storing this kind of data. The model is applied to the Lizzie Research Site, North Carolina, in which the total nitrogen loading to surface waters via ground-water transport is calculated in response to the spreading on the (land) surface of feedlot waste. The measurements agree with predictions, but the authors point out that this could be coincidental due to the high level of heterogeneity of the domain of calculation.

Martinez *et al.* (2008) extend an existing hydrologic model that is written in procedural language in an object-oriented way using Java. The extensions included the representation of the water-table, the effects of (plant and soil) evapotranspiration on a shallow water-table, and the influence of time-varying surface water levels. The model is validated by comparing model predictions with measurements from two experimental sites in southeastern USA.

2.6 Summary

This chapter has provided an introduction to the different aspects of the subject of modelling natural flows using analytical, numerical and management techniques.

We have defined the mathematical problem that will be solved, including what form the source terms will take. We have briefly mentioned the special case of shallow water in circular domains.

Different types of numerical schemes have been described: the advantage of the FVM in naturally conserving mass and the unstructured triangular mesh's ability in dealing with the irregular boundaries of a two-dimensional natural domain justify their use in the *Riemann2D* package.

Applications of the SWEs to natural bays and lakes using numerical schemes have been presented. Many of the test domains are smaller than the ones used in this thesis. Some use tidal data as boundary conditions. All tests are verified by comparison with either experimental data, field measurements, or analytical solutions. In particular, FVM schemes on unstructured grids with a limiter show good agreement on domains up to 20 km across.

Several simulations of pollutant transport have been given. Many used an advection equation to model the pollutant as passive, which is the approach we will take. The most important result from this section is that all the models and measurements find that wind is the most effective mechanism in transportation, either of sediment or of pollutant.

Some issues surrounding the importance of water to Brisbane have been discussed. Although the work in this thesis is not about the supply of water to Brisbane, the literature highlights the importance of any type of water modelling because water is such a valuable and scarce resource in that part of the world. A number of field studies have been carried out in Moreton Bay. Some of these measurements will be used to calibrate results in later chapters.

Finally, we have given a brief outline of object orientation and Java as this is the language used to write *Riemann2D* . Many studies of water modelling that use the OO approach are interdisciplinary and allow for different management practices to be modelled, as well as the physical and biological effects.

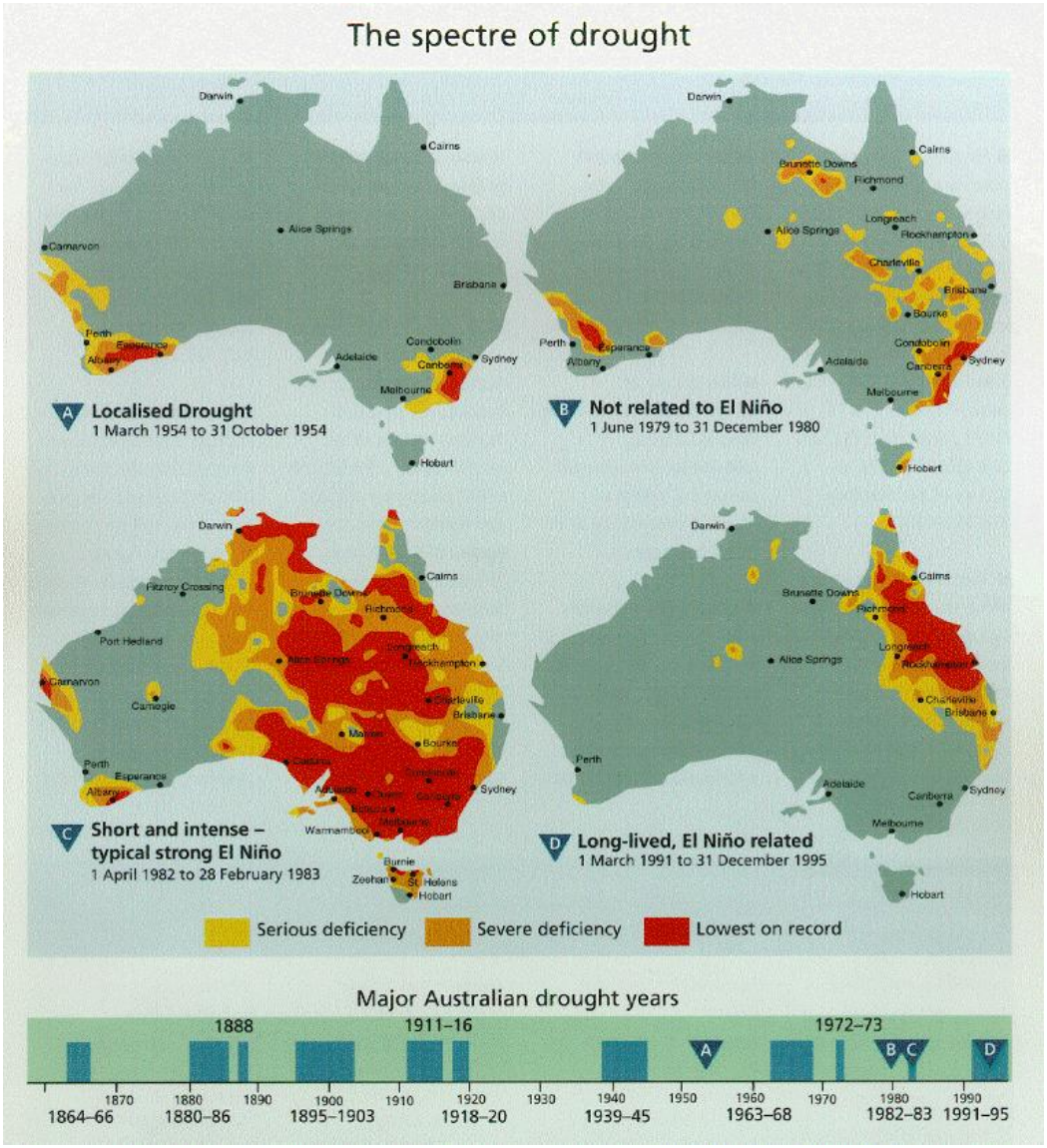


Figure 2.3: Reproduced from the Australian Government's Bureau of meteorology website `www.bom.gov.au/climate/drought/livedrought.shtml`. Maps and figures showing areas of Australia affected by drought.

CHAPTER 3

Mathematical properties

The shallow water equations are a well-known system of nonlinear, hyperbolic PDEs, that describe the advective flow in the horizontal plane, where vertical motion can be neglected. The shallow water equations can be derived from the Navier-Stokes equations by depth integration under the assumption that the vertical length scale is much smaller than the horizontal and can thus be ignored. They are a classical system with a large amount of literature, both analytical and numerical.

The first part of this chapter concerns classical analysis, including eigenvalues, Riemann invariants, the Riemann problem and the idea of weak solutions. In the second part, asymptotic analysis is carried out for the particular case when the problem time is large and the flow velocity is small. The relative importance of source terms is studied. The asymptotic solutions given here not only provide insight into the behaviour of the system but allow for verification of results in later chapters.

3.1 Classical analysis

First the definitions of some terms used in this chapter are given.

Hyperbolicity

Hyperbolic problems are distinguished from elliptic or parabolic problems by the following properties: the eigenvalues are real and distinct and there is an inherent directional property of the solutions. The solutions are wave-like in nature (this is also true of parabolic PDEs), and in particular a system of m equations should give rise to m distinct waves (LeVeque (2003)). A matrix associated with the hyperbolic problem is sometimes diagonalisable, and this property may allow the system to be decoupled and hence solved more simply.

The shallow water equations are hyperbolic provided $h > 0$. When the depth of water becomes vanishingly small, the regime is then elliptic. (The transition of a system from the hyperbolic to the elliptic domain has been studied by Milewski *et al.* (2004) and LeVeque (2003).)

Linearity

Definition For a PDE of the form

$$a(x,y,u)\frac{\partial u}{\partial x} + b(x,y,u)\frac{\partial u}{\partial y} = \kappa(x,y,u), \tag{3.1}$$

we say that (3.1) is **linear** if a and b are independent of u, and κ can be written as a linear function of u, for example $\kappa(x,y,u) = \alpha(x,y)u + \beta(x,y)$ for some constants α, β. If κ cannot be written like this, we say (3.1) is **semilinear**. In the case where a, b and κ all depend on u (but, importantly, not on the *derivatives* of u), we say (3.1) is **quasilinear** (see, for example Ockendon *et al.* (1999)). The shallow water equations are therefore quasilinear.

Conservation form

Definition A system of PDEs in the form

$$\frac{\partial S}{\partial t} + \frac{\partial P}{\partial x} + \frac{\partial Q}{\partial y} = R, \tag{3.2}$$

for some vectors S, P, Q and R all functions of h, u and v (or other relevant variables), is said to be in *conservation form*.

This means the variables all appear in their conserved form, so that the quantities are advected with the flow and mass is naturally conserved. When the FVM formulation is used

(see Chapter 4), it is essential that the equations are in this form, in order to accurately model the propagation of shocks, should they arise in the flow.

If an equation in conservation form can be written

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial f(\mathbf{q})}{\partial x} + \frac{\partial g(\mathbf{q})}{\partial y} = 0,$$

it is said to be an advection equation for $\mathbf{q}$. If the flux function $f(\mathbf{q})$ is linear, the solution is often easy to obtain, and information is advected along with the flow. However, if f is a nonlinear vector function of $\mathbf{q}$, the solution is able to evolve with time in much more complicated ways. The information propagates as a function of the solution, and so it is possible that discontinuous solutions develop over time even if initial data is smooth (LeVeque (2003)).

3.1.1 Eigenvalues

Recall the shallow water equations (2.2) from Chapter 2 in conservation form

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{s},$$

where

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} q_2 \\ q_2^2/q_1 + gq_1^2/2 \\ q_2 q_3/q_1 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} q_3 \\ q_2 q_3/q_1 \\ q_3/q_1 + gq_1^2/2 \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 0 \\ -gq_1 b_x \\ -gq_1 b_y \end{bmatrix}. \tag{3.3}$$

The vector $\mathbf{q} = [h,\, hu,\, hv]^{\mathrm{T}}$ is the vector of conserved variables.

Note, if the non-conserved variables, $[h,\, u,\, v]^{\mathrm{T}}$, are used, the system takes the form (2.3).

To calculate the eigenvalues of the shallow water equations, define the Jacobian matrix of differentiated coefficients of $\mathbf{F}(\mathbf{q}) = [f_1,\, f_2,\, f_3]^{\mathrm{T}}$ as $\mathbf{F}'(\mathbf{q}) = (\partial f_i/\partial q_j)$ for $i, j = 1, 2, 3$:

$$\mathbf{F}'(\mathbf{q}) = \begin{bmatrix} \dfrac{\partial f_1}{\partial q_1} & \dfrac{\partial f_1}{\partial q_2} & \dfrac{\partial f_1}{\partial q_3} \\ \dfrac{\partial f_2}{\partial q_1} & \dfrac{\partial f_2}{\partial q_2} & \dfrac{\partial f_2}{\partial q_3} \\ \dfrac{\partial f_3}{\partial q_1} & \dfrac{\partial f_3}{\partial q_2} & \dfrac{\partial f_3}{\partial q_3} \end{bmatrix},$$

and an equivalent expression for $\mathbf{G}'$, then (3.3) in matrix form is

$$\mathbf{q}_t + \mathbf{F}'(\mathbf{q})\frac{\partial \mathbf{q}}{\partial x} + \mathbf{G}'(\mathbf{q})\frac{\partial \mathbf{q}}{\partial y} = \mathbf{s}, \tag{3.4}$$

where

$$\mathbf{F}'(\mathbf{q}) = \begin{bmatrix} 0 & 1 & 0 \\ gq_1 - q_2^2/q_1^2 & 2q_2/q_1 & 0 \\ -q_2q_3/q_1 & q_3/q_1 & q_2/q_1 \end{bmatrix}, \quad \mathbf{G}'(\mathbf{q}) = \begin{bmatrix} 0 & 0 & 1 \\ -q_2q_3/q_1 & q_3/q_1 & q_2/q_1 \\ gq_1 - q_3^2/q_1^2 & 0 & 2q_3/q_1 \end{bmatrix}.$$

Because this system has two spatial dimensions, the waves (solutions) move in some arbitrary direction given by the unit normal vector $\mathbf{n} = (n_x,\ n_y)$. Following, for example, Brufau and Garcia-Navarro (2000) or LeVeque (2003), the eigenvalues of the entire system are found by solving $|n_x\mathbf{F}' + n_y\mathbf{G}' - \lambda_i I| = 0$, for $i = 1, 2, 3$, which yields

$$\lambda_1 = n_xq_2/q_1 + n_yq_3/q_1 - \sqrt{gq_1}, \quad \lambda_2 = n_xq_2/q_1 + n_yq_3/q_1, \quad \lambda_3 = n_xq_2/q_1 + n_yq_3/q_1 + \sqrt{gq_1},$$

or reverting to the original variables and using $\mathbf{n} \cdot \mathbf{u} = (n_x,\ n_y) \cdot (u,\ v)$

$$\lambda_1 = \mathbf{n} \cdot \mathbf{u} - c, \quad \lambda_2 = \mathbf{n} \cdot \mathbf{u}, \quad \lambda_3 = \mathbf{n} \cdot \mathbf{u} + c, \tag{3.5}$$

where $c = \sqrt{gh}$ is the wavespeed. Since these are real and distinct for $h > 0$, the system is hyperbolic.

Froude number

The eigenvalues take the form of a convective velocity minus/plus a phase velocity. Dividing convective by phase yields a Froude number for each x- and y-direction:

$$\frac{u^2}{gh} = \mathrm{Fr}^2_{(x)}, \quad \text{and} \quad \frac{v^2}{gh} = \mathrm{Fr}^2_{(y)}. \tag{3.6}$$

The Froude number may be thought of as a relation of inertia to gravitational forces: the top term being related to kinetic energy, the bottom to potential energy. Flows with Froude numbers less than one are said to be *subcritical*, and flows with Froude number greater than one are said to be *supercritical*. Examples of supercritical flows include a sonic boom, a dam break, or a bore; flow velocity is high compared to the characteristic flow depth. In this thesis, we are considering slow flows where inertia is small and gravity dominates, thus we expect the Froude number to be small.

3.1.2 Riemann invariants

The existence of Riemann invariants is unusual for a PDE, but the SWEs is one system for which these invariants do exist. As a result, the system can be written in its characteristic

form, as has been done by Alcrudo and Benkhaldoun (2001), Karelsky and Petrosyan (2006) and Li and Chen (2006), among many others. In their characteristic form, the equations may decouple to an integrable form and so an analytical solution could be found.

Definition If it is possible to write a PDE in the form

$$\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\tau} = 0,$$

for some function $\mathcal{R}(\tau)$ with parameter τ along a curve $\mathcal{C}(\tau)$, then we say that $\mathcal{R}$ is *Riemann invariant* along $\mathcal{C}(\tau)$ for that PDE. Therefore, if a PDE can be arranged into the form

$$\mathbf{l}_i^{\mathrm{T}} A \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\tau} = \mathbf{l}_i^{\mathrm{T}} \mathbf{s}, \tag{3.7}$$

(where $\mathbf{l}_i^{\mathrm{T}}$ is a left eigenvector of A, the flux matrix, λ_i the eigenvalue that is the slope $\mathrm{d}y/\mathrm{d}x$ of $\mathcal{C}$, τ is a parameter and $\mathbf{s}$ is the vector of the right hand side) and it is possible to integrate this, then this is equivalent to finding such an $\mathcal{R}$.

Given initial data for the problem, *e.g.* $\mathbf{q}(x, y, 0) = \mathbf{q}_0(x, y)$ along an initial data curve, Riemann invariants may be evaluated along this curve. But since each of these Riemann invariants is conserved along its respective characteristic, this is equivalent to knowing Riemann invariants for a range of x and t. This in turn is equivalent to knowing the solution $\mathbf{q}(x, y, t)$ in the domain of definition from the initial data. Because the Riemann invariants are weakly coupled to the eigenvalues, the characteristics are not necessarily straight lines (Tabak (2007)).

The system (3.3) may or may not have explicit analytical solutions, depending on the geometry of the bay, the initial data, and the source terms.

Although the existence of Riemann invariants means there is a better chance of finding an analytical solution, it is by no means a guarantee. Indeed, analytical solutions usually cannot be found if the system contains a shock. However, (see Ockendon *et al.* (1999)) if initial or boundary data is given, such that one Riemann invariant is constant everywhere and not just on characteristics, it may be possible to integrate and obtain a solution.

More information on Riemann invariants can be found in the Appendix B and also in Ockendon *et al.* (1999).

Riemann invariants for the 2D shallow water equations

Consider the shallow water system given in (3.4)

$$\frac{\partial \mathbf{q}}{\partial t} + \mathbf{F}' \frac{\partial \mathbf{q}}{\partial x} + \mathbf{G}' \frac{\partial \mathbf{q}}{\partial y} = \mathbf{s}, \tag{3.8}$$

where

$$\mathbf{q} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{F}' = \begin{bmatrix} 0 & 1 & 0 \\ gh - u^2 & 2u & 0 \\ -uv & v & u \end{bmatrix}, \quad \mathbf{G}' = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ gh - v^2 & 0 & 2v \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 0 \\ -ghb_x \\ -ghb_y \end{bmatrix}.$$

Define as before the unit normal vector $\mathbf{n} = (n_x, n_y)$. Then (3.8) is written

$$\frac{\partial \mathbf{q}}{\partial t} + (n_x \mathbf{F}' + n_y \mathbf{G}') \frac{\partial \mathbf{q}}{\partial n} = \mathbf{s},$$

where $\partial \mathbf{q}/\partial n$ is defined via

$$\mathbf{n} \cdot \frac{\partial \mathbf{q}}{\partial n} = \frac{\partial \mathbf{q}}{\partial x} + \frac{\partial \mathbf{q}}{\partial y} = \nabla \cdot \mathbf{q}.$$

The eigenvalues of this system are given in (3.5). The left eigenvectors for (3.8) are the $\mathbf{l}_i^{\mathrm{T}}$ that satisfy

$$\mathbf{l}_i^{\mathrm{T}} (n_x \mathbf{F}' + n_y \mathbf{G}') = \mathbf{l}_i^{\mathrm{T}} \lambda_i, \tag{3.9}$$

which are

$$\begin{array}{ccc} \lambda_1 = \mathbf{n} \cdot \mathbf{u} - c & \lambda_2 = \mathbf{n} \cdot \mathbf{u} & \lambda_3 = \mathbf{n} \cdot \mathbf{u} + c \\ \mathbf{l}_1 = \begin{bmatrix} -\mathbf{n} \cdot \mathbf{u} - c & n_x & n_y \end{bmatrix} & \mathbf{l}_2 = \begin{bmatrix} n_y u - n_x v & -n_y & n_x \end{bmatrix} & \mathbf{l}_3 = \begin{bmatrix} -\mathbf{n} \cdot \mathbf{u} + c & n_x & n_y \end{bmatrix}. \end{array}$$

Applying these in the formula for Riemann invariants (3.7):

i) $\lambda_{1,3} = \mathbf{n} \cdot \mathbf{u} \pm c$ Left-multiply the system (3.8) by $\mathbf{l}_{1,3}^{\mathrm{T}}$. After rearranging, the Riemann invariants for the 1- and 3-families are found to be

$$\mathbf{n} \cdot \mathbf{u} + \mathbf{n} \cdot gt(b_y, b_x) \pm 2c = \text{const} = \mathcal{R}_{1,3}.$$

Thus $\mathbf{n} \cdot \mathbf{u} + \mathbf{n} \cdot gt(b_y, b_x) \pm 2c$ is constant along characteristics $\mathcal{C}_{1,3}$ where $(\mathrm{d}y/\mathrm{d}x)_{1,3} = \lambda_{1,3} = \mathbf{n} \cdot \mathbf{u} \pm c$ respectively.

ii) $\lambda_2 = \mathbf{n} \cdot \mathbf{u}$ Left-multiply the system (3.8) by $\mathbf{l}_2^{\mathrm{T}}$, and again rearrange to find the Riemann invariant for the 2-family

$$\mathbf{n} \cdot (v + gb_y t,\ -u - gb_x t) = \text{const} = \mathcal{R}_2.$$

Thus $\mathbf{n} \cdot (v + gb_y t,\ -u - gb_x t)$ is constant along characteristic $\mathcal{C}_2 = (\mathrm{d}y/\mathrm{d}x)_2 = \lambda_2 = \mathbf{n} \cdot \mathbf{u}$.

It is well known that the Riemann invariants for the one-dimensional shallow water equations in a channel with a horizontal bottom are $u \pm 2\sqrt{gh}$ (see, for example Ockendon *et al.* (1999)). The above results match with the classical theory: in the case of non-varying bottom topography, *i.e.* $b_x = b_y = 0$, the Riemann invariants are $\mathcal{R}_{1,3} = \mathbf{n} \cdot \mathbf{u} \pm 2\sqrt{gh}$ along $\lambda_{1,3} = \mathbf{n} \cdot \mathbf{u} \pm \sqrt{gh}$ and $\mathcal{R}_2 = \mathbf{n} \cdot (v,\ -u) = \mathbf{n}^{-1} \cdot \mathbf{u}$ along $\lambda_2 = \mathbf{n} \cdot \mathbf{u}$ (where $\mathbf{n}^{-1} = (n_y,\ -n_x)$).

In his book, LeVeque (2003) employs a method for finding the Riemann invariants that uses the right eigenvectors. The resulting invariants are identical to those of the left eigenvectors except for a change of sign. This change of sign can be explained by the fact that the left and right eigenvectors are perpendicular. Since the right Riemann invariants point in a different direction from the left, the sign is necessarily different.

The Riemann invariants of the left and right eigenvectors are displayed in Table 3.1 for easy comparison.

	left	right
1-family	$(1,\ 1) \cdot (\mathbf{n} \cdot \mathbf{u},\ 2c)$	$(\mathbf{n},\ -\mathbf{n}) \cdot (\mathbf{n} \cdot \mathbf{u},\ 2c)$
2-family	$(1,\ 0) \cdot (\mathbf{n}^{-1} \cdot \mathbf{u},\ 1)$	$(\mathbf{n},\ \mathbf{n}^{-1}) \cdot (\mathbf{n}^{-1} \cdot \mathbf{u},\ 1)$
3-family	$(1,\ 1) \cdot (\mathbf{n} \cdot \mathbf{u},\ -2c)$	$(\mathbf{n},\ -\mathbf{n}) \cdot (\mathbf{n} \cdot \mathbf{u},\ -2c)$

Table 3.1: Table showing relationship between Riemann invariants for left and right eigenvectors.

3.1.3 The Riemann problem

The Riemann problem, as noted in Section 2.2.2, is a hyperbolic problem together with some initial data that is discontinuous at a point in space, *i.e.*

$$\begin{aligned} \mathbf{q}_t + f(\mathbf{q})_x &= 0 \\ \mathbf{q}(0,x) &= \begin{cases} q_L & \text{if } x > 0, \\ q_R & \text{if } x < 0. \end{cases} \end{aligned} \tag{3.10}$$

Depending on the nature of f, solutions of the system (3.10) may be obtained. If the function f is linear, we expect the discontinuity to propagate at a speed that is an eigenvalue of f. In this case, the solution can be found explicitly (LeVeque (2003)). If on the other hand f is nonlinear, more work is required to find a solution.

Physically, the Riemann problem can be interpreted as describing a readily-observed phenomenon: the Severn bore [1], the supersonic boom of an aeroplane flying past, a dam break, or the effect of a kitchen tap being turned on fast, are all examples of a shock that is described by (3.10). Because of the wide range of problems that this system can be applied to, it is a very well-researched area (see Roe (1981), Glaister (1988a), Glaister (1988b), Bermúdez and Vázquez (1994) for numerical schemes for gas dynamics and the SWEs, Alcrudo and Benkhaldoun (2001), Karelsky and Petrosyan (2006), Li and Chen (2006) for theoretical analysis of the Riemann problem of the shallow water equations with an irregular bottom topography, and some classical texts include Whitham (1974), Toro (1999), Ockendon *et al.* (1999)).

The Riemann problem is the basis for a large class of numerical schemes that are essentially a series of local Riemann problems, which are known as *exact* (*e.g.* due to Godunov) or *approximate Riemann solvers* (perhaps most famously due to Roe (1981)). Some others due to van Leer (1977), Osher and Solomon (1982) and Harten (1983) have been developed by Jha (2006) in the context of *Riemann2D* .

1. Running from Ceredigion in Wales to the Severn estuary and the Bristol Channel, the river Severn runs through Powys, Shropshire, Worcestershire and Gloucestershire. It is the longest river in Britain, at 354 km, and when the tide in the Bristol Channel comes in, the effect is to create a discontinuity in the surface of the water that travels upstream as far as Gloucester. The visible bores (ones in which total depth is greater than 4.5 metres) happen around 50 times a year, and are greatest in the spring. See the Environment Agency's website `www.environment-agency.gov.uk/regions/midlands/434823/?lang=_e` for more details

A discontinuity at a point in 1D is a discontinuity on a line in 2D. Thus to formulate the Riemann problem in two dimensions, we consider the state variable $\mathbf{q}(0, x, y)$ to have the value $\mathbf{q}_L$ to one side of the line and $\mathbf{q}_R$ to the other. The definition of the line of discontinuity is given in Sections 4.1.2 and 4.1.4.

3.1.4 Rankine–Hugoniot conditions and weak solutions

The idea of weak solutions was first proposed by Jean Leray in 1934 (Borel *et al.* (2000)). A weak solution to a PDE is a solution that satisfies the PDE in some specified sense, but for which derivatives do not in general exist. A PDE that admits discontinuities in its solution can still be formulated in a rigourous way, which is useful when dealing with a Riemann problem (Ockendon *et al.* (1999)).

Consider a volume, V, bounded by two curves Γ, across which the solution is discontinuous, and another arbitrary curve γ. Suppose the PDE in its conservation form is

$$\frac{\partial P}{\partial t} + \frac{\partial Q}{\partial x} + \frac{\partial R}{\partial y} = S. \tag{3.11}$$

Multiplying (3.11) by a test function [2], ψ, and using the chain rule we have

$$\begin{aligned} & \psi\frac{\partial P}{\partial t} + \psi\frac{\partial Q}{\partial x} + \psi\frac{\partial R}{\partial y} = \psi S \\ \Rightarrow \quad & \frac{\partial}{\partial t}(\psi P) + \frac{\partial}{\partial x}(\psi Q) + \frac{\partial}{\partial y}(\psi R) = P\frac{\partial \psi}{\partial t} + Q\frac{\partial \psi}{\partial x} + \frac{\partial}{\partial y}(\psi R) + \psi S. \end{aligned}$$

Define the divergence to be over time and space, *i.e.*

$$\nabla \cdot \; = \frac{\partial}{\partial t} + \frac{\partial}{\partial x} + \frac{\partial}{\partial y},$$

and let the vector $\mathbf{P} = (P, Q, R)$. Now integrate over the region V,

$$\iiint_V \nabla \cdot (\psi \mathbf{P}) \, \mathrm{d}V = \iiint_V P\frac{\partial \psi}{\partial t} + Q\frac{\partial \psi}{\partial x} + R\frac{\partial \psi}{\partial y} + \psi S \, \mathrm{d}V.$$

Apply the divergence theorem, and since ψ is arbitrary, it can be chosen so that it vanishes on the arbitrary curve γ

$$\int_\Gamma \psi \mathbf{P} \cdot \mathbf{n} \, \mathrm{d}s = \iiint_V P\frac{\partial \psi}{\partial t} + Q\frac{\partial \psi}{\partial x} + R\frac{\partial \psi}{\partial y} + \psi S \, \mathrm{d}V, \tag{3.12}$$

2. Stakgold (1979) gives the following definition: "A *test function* $\psi(x) = \psi(x_1, \ldots, x_n)$ on R_n is a function which is infinitely differentiable on R_n and vanishes outside some bounded region". Test functions are often used in physics and engineering to allow the treatment of equations that admit weak solutions.

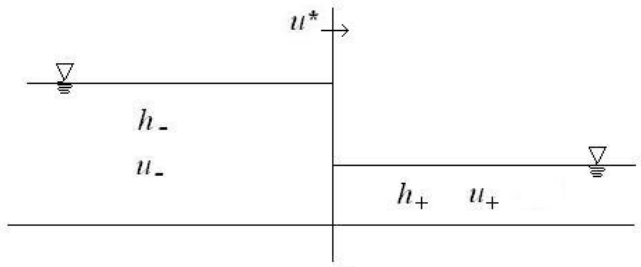


Figure 3.1: Shallow water with a discontinuity of depth moving at speed u^*.

where s is a parameterisation of the curve Γ. Now P, Q and R represent solutions of (3.11), but their derivatives do not appear in (3.12). Although a solution satisfying (3.11) also satisfies (3.12), it is possible that there are solutions that satisfy (3.12) but whose derivatives do not exist, hence they fail to satisfy (3.11).

Since many applied problems do not admit sufficiently smooth solutions, it is sometimes necessary to work with the weak formulation of the problem (3.12). Solving the Riemann problem means treating the state variables as initially constant in each region, and as discontinuous across the interface of these regions. Thus the weak formulation will be the formulation to use.

Rankine-Hugoniot conditions

The Rankine–Hugoniot conditions are a tool that determines one unknown of the system, when all but one of the variables are known.

Consider a 1D section of shallow water with a discontinuity moving with speed u^* that has water depth h_- in $x < u^*t$ and h_+ in $x > u^*t$ such that $h_- > h_+$, as shown in Figure 3.1. The water behind the discontinuity is moving with speed u_- and ahead it is stationary, so $u_+ = 0$. The mass crosses the shock from the left at a rate $h_-(u_- - u^*)$ and leaves it on the right at a rate $h_+(u_+ - u^*)$. Since mass must be conserved across the shock, we have

$$[h(u - u^*)]_-^+ = 0,$$

where $[\cdot]_-^+$ denotes the change from the left side $(-)$ to the right $(+)$ (see Ockendon and

Ockendon (2004)). By Newton's second law, force equals rate of change of momentum thus

$$[h(u-u^*)^2 + \tfrac{1}{2}gh^2]_-^+ = 0.$$

Note that energy is not conserved across the shock. Since energy cannot be created, it can only be dissipated across the shock, which is equivalent to saying that the water depth increases as the shock passes. Now we solve the Rankine–Hugoniot conditions to determine u^*

$$h_-(u_- - u^*) = -h_+u^*$$
$$h_-(u_- - u^*)^2 + \tfrac{1}{2}gh_-^2 = h_+u^{*2} + \tfrac{1}{2}gh_+^2.$$

Using the condition that mass is conserved allows u_- to be eliminated,

$$\frac{h_+}{h_-}(h_+ - h_-)u^{*2} = \tfrac{1}{2}g(h_+^2 - h_-^2)$$

thus u^* is determined

$$u^* = \pm\sqrt{\frac{gh_+(h_- + h_+)}{2h_-}},$$

where we choose the root that gives $u^* > 0$ so that $h_- > h_+$ (*i.e.* to satisfy the energy condition).

3.1.5 Energy

We have taken the conservation of mass and momentum of the system as our basic principles. We now consider the consequence of these assumptions on the conservation, or otherwise, of energy. Because there are three unknowns, h, u, v, and three equations, we do not expect any extra constraint on the unknowns, since the system might then become unsolvable.

Note that only mechanical energy, and not thermal, is considered here.

The energy of the system is the sum of kinetic and potential energies, thus energy, E, is

$$E = \int_A \frac{1}{2}\rho h(\mathbf{u}\cdot\mathbf{u}) + \rho\frac{1}{2}gh^2 \,\mathrm{d}A, \tag{3.13}$$

for some area A. Now consider its time derivative, using equations (2.3) to substitute for h_t,

$\mathbf{u}_t$:

$$\begin{aligned}
\frac{\mathrm{d}E}{\mathrm{d}t} &= \rho \int_A \frac{1}{2}(\mathbf{u}\cdot\mathbf{u})h_t + (\mathbf{u}\cdot\mathbf{u}_t)h + ghh_t \,\mathrm{d}A, \\
&= -\rho \int_A \left(\frac{1}{2}(\mathbf{u}\cdot\mathbf{u}) + gh\right)(\mathbf{u}\cdot\nabla h + h\nabla\cdot\mathbf{u}) + \mathbf{u}\cdot h\left((\mathbf{u}\cdot\nabla)\mathbf{u} + g\nabla h - \hat{\mathbf{s}}\right)\,\mathrm{d}A, \\
&= -\rho \int_A \nabla\cdot\left(\left(\frac{1}{2}(\mathbf{u}\cdot\mathbf{u}) + gh\right)\mathbf{u}h\right)\,\mathrm{d}A + \rho\int_V \mathbf{u}\cdot h\hat{\mathbf{s}}\,\mathrm{d}A, \qquad (3.14) \\
&= -\rho \int_{\partial A} \left(\frac{1}{2}(\mathbf{u}\cdot\mathbf{u}) + gh\right)\cancelto{0}{\mathbf{u}h\cdot\mathbf{n}}\,\mathrm{d}S + \rho\int_A \mathbf{u}\cdot h\hat{\mathbf{s}}\,\mathrm{d}A. \qquad (3.15)
\end{aligned}$$

By applying the divergence theorem at (3.14) and using the fact that there is no water flux (in 3.15) on the boundary, *i.e.* that $h\mathbf{u}\cdot\mathbf{n} = 0$ on ∂A, we have shown that energy is added to the system only through $\hat{\mathbf{s}}$, the right hand side terms that include inflows, outflows, bed friction, wind stress and other source terms. That is, the work done by the right hand side force terms in (2.3b) increases or decreases the energy of the system. If there is tidal inflow on a part of the boundary, then $h\mathbf{u}\cdot\mathbf{n}$ equals the water influx, and this also changes the energy. This result provides insight into what to expect from the numerical simulations. The amount of energy added to the system depends only on the boundary conditions and the source terms.

3.2 Asymptotics

Asymptotic analysis is a way of investigating the limiting behaviour of a system. First the size of each term is determined by nondimensionalisation then smaller terms are neglected to obtain a simplified system that can be solved. Then progressively smaller terms are brought back in to increase the accuracy of the solution. As the number of smaller terms added tends to infinity, the solution tends to the exact solution. For practical purposes, three orders of magnitude is considered acceptable.

3.2.1 Nondimensionalisation

In nondimensionalisation, each variable is transformed to a unit variable and a relevant coefficient. The choice of the coefficient for each variable is almost never obvious. It requires more intuition than pure science, hence is often called an art (Fowler (1997)). A successful choice yields one (or more) nondimensional parameter about which to expand the variables and hence approximate a solution.

We perform the nondimensionalisation by writing $h = D\tilde{h}$, where D is the size of the variable and $\tilde{h}$ is the nondimensional form of the variable. Let b scale in the same way, that is $b = D\tilde{b}$. Further, let u, v scale with U (ms^{-1}), x, y scale with L (m), and t with LU^{-1} (s). We have

$$h = D\tilde{h}, \quad u = U\tilde{u}, \quad v = U\tilde{v}, \quad t = LU^{-1}\tilde{t}, \quad x = L\tilde{x}, \quad y = L\tilde{y}, \quad b = D\tilde{b}. \tag{3.16}$$

We consider a bay in which the depth of the water in the bay is $D = \mathcal{O}(10)\,\mathrm{m}$. From the literature review in Chapter 2, we take the velocity of the water to be small, so that $U = \mathcal{O}(10^{-1})\,\mathrm{ms}^{-1}$, and we have the usual gravitational constant $g = \mathcal{O}(10)\,\mathrm{ms}^{-2}$. The Froude number (recall (3.6)) is therefore $\mathrm{Fr}^2 = \mathcal{O}(10^{-4})$.

Assume the bay has typical length $L = \mathcal{O}(10^5)\,\mathrm{m}$ and since $U = \mathcal{O}(10^{-1})\,\mathrm{ms}^{-1}$, the timescale of the problem is $\mathcal{O}(10^4)\,\mathrm{s}$, or about 3 hours.

Substituting the variables (3.16) into (2.1) and dropping the tildes immediately, the nondimensional form is

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = 0 \tag{3.17a}$$

$$\mathrm{Fr}^2 \left(\frac{\partial (hu)}{\partial t} + \frac{\partial (hu^2)}{\partial x} + \frac{\partial (huv)}{\partial y} \right) + h\frac{\partial h}{\partial x} = -h\frac{\partial b}{\partial x} \tag{3.17b}$$

$$\mathrm{Fr}^2 \left(\frac{\partial (hv)}{\partial t} + \frac{\partial (huv)}{\partial x} + \frac{\partial (hv^2)}{\partial y} \right) + h\frac{\partial h}{\partial y} = -h\frac{\partial b}{\partial y}. \tag{3.17c}$$

The nondimensional parameter of this system is the Froude number

$$\mathrm{Fr}^2 = \frac{U^2}{gD},$$

which multiplies the advective terms. Since it is small this means the water depth is principally determined by the right hand side *i.e.* by the bathymetry of the domain. Notice that the conservation of mass equation is already in nondimensional form.

We denote by ε the small parameter and write $\varepsilon^2 = \mathrm{Fr} = \mathcal{O}(10^{-2})$ and expand the variables in terms of the small parameter ε as

$$h = h_0 + \varepsilon h_1 + \varepsilon^2 h_2 + \varepsilon^3 h_3 + \mathcal{O}(\varepsilon^4)$$

$$p = p_0 + \varepsilon p_1 + \varepsilon^2 p_2 + \varepsilon^3 p_3 + \mathcal{O}(\varepsilon^4)$$

$$q = q_0 + \varepsilon q_1 + \varepsilon^2 q_2 + \varepsilon^3 q_3 + \mathcal{O}(\varepsilon^4).$$

Let the vector of flux variables [1] be $\mathbf{p} = (p,\, q) = (uh,\, vh)$. Using vector notation, the mass equation (3.17a) becomes

$$\varepsilon^0 \left(h_{0t} + \nabla \cdot \mathbf{p_0}\right) + \varepsilon \left(h_{1t} + \nabla \cdot \mathbf{p}_1\right) + \varepsilon^2 \left(h_{2t} + \nabla \cdot \mathbf{p}_2\right) + \varepsilon^3 \left(h_{3t} + \nabla \cdot \mathbf{p}_3\right) + \mathcal{O}(\varepsilon^4) = 0, \quad (3.18)$$

and the momentum equations ((3.17b)-(3.17c)) are

$$\varepsilon^0 \left(h_0 \nabla(h_0 + b)\right) + \varepsilon \left(h_0 \nabla h_1 + h_1 \nabla(h_0 + b)\right) + \varepsilon^2 \left(\nabla(h_0 h_2) + h_1 \nabla h_1 + h_2 \nabla b\right) + \varepsilon^3 \left(\nabla(h_0 h_3) + \nabla(h_1 h_2) + h_3 \nabla b\right) + \mathcal{O}(\varepsilon^4) = 0. \quad (3.19)$$

3.2.2 Boundary conditions

Consider a domain D that is bounded by $\partial D_1 \cup \partial D_2$, where ∂D_1 represents the 'wall' of the bay, and ∂D_2 represents the ocean boundary of the bay. Let there be zero normal flux on the wall, ∂D_1, thus

$$\mathbf{p}(x, y, t) \cdot \mathbf{n} = 0 \quad \text{for} \quad x, y \in \partial D_1. \quad (3.20)$$

We also have, on the ∂D_1 boundary

$$h(x, y, t) \cdot \mathbf{n} = \mathbf{0} \quad \text{for} \quad x, y \in \partial D_1. \quad (3.21)$$

Flux through the ocean boundary, ∂D_2, is specified: the water level is governed by the level of the still water plus the effect of the tide, so

$$h(x, y, t) = -b(x, y) + \varepsilon \phi_1(t) \quad \text{for} \quad x, y \in \partial D_2, \quad (3.22)$$

where $\phi_1(t)$ is a term representing the effect of the tide, which depends only on time, not on x or y (in Chapter 7 there is more discussion on the choice of $\phi_1(t)$). The flux across this boundary is

$$\mathbf{p} \cdot \mathbf{n} = \mathbf{p}_0 \cdot \mathbf{n} + \varepsilon \left(\mathbf{u}_0 h_1 + \mathbf{u}_1 h_0\right) \cdot \mathbf{n} + \varepsilon^2 \left(\mathbf{u}_0 h_2 + \mathbf{p}_1 + \mathbf{u}_2 h_0\right) \cdot \mathbf{n} + \varepsilon^3 \left(\mathbf{u}_0 h_3 + \mathbf{u}_1 h_2 + \mathbf{u}_2 h_1 + \mathbf{u}_3 h_0\right) \cdot \mathbf{n} + \ldots$$

1. For notational purposes, to avoid the over-use of subscripts, we use $(p,\, q) = (uh,\, vh)$ rather than the more conventional $(q_2,\, q_3) = (uh,\, vh)$ in this section.

$$= k_0(x,y,t) + \varepsilon k_1(x,y,t) + \varepsilon^2 k_2(x,y,t) + \dots, \quad \text{for} \quad x,y \in \partial D_2 \text{ and for all } t, \tag{3.23}$$

where k is a flux function. Consider the flux functions k_0, k_1, k_2, k_3. On the ocean boundary at leading order, $h_0(x,y,t) = -b(x,y)$ thus $k_0 = -\mathbf{u}_0 b$. At order $\mathcal{O}(\varepsilon)$, the water depth is $\phi_1(t)$, so $k_1 = -\mathbf{u}_1 b + \mathbf{u}_0 \phi_1$. At $\mathcal{O}(\varepsilon^2)$, assuming u_2, v_2 and h_2 are all zero on ∂D_2, we have $k_2 = \mathbf{u}_1 \phi_1(t)$. Assuming u_3, v_3 and h_3 are all zero on ∂D_2, we have $k_3 = 0$.

Leading order

The leading order momentum equation (3.19) is

$$\begin{aligned} & h_0 \nabla(h_0 + b) = 0 \\ \text{for } h_0 \neq 0 \qquad & \nabla(h_0 + b) = 0 \\ \Rightarrow \qquad & h_0 = -b(x,y) + \phi_0(t). \end{aligned}$$

But by condition (3.22) at ∂D_2, we have $\phi_0 = 0$ thus $h_0 = -b(x,y)$.

Use this result in the equation for mass (3.18): since h_0 is not a function of t, the first term is zero, leaving

$$\nabla \cdot \mathbf{p}_0 = 0.$$

Let $(p_0, q_0) = \nabla A_0$ for some function $A_0 = A_0(x,y,t)$, so that $\nabla^2 A_0 = 0$, which is Laplace's equation. Note that condition (3.20) can be written equivalently as $\nabla A \cdot \mathbf{n} = 0$, or $\partial A / \partial n = 0$ on ∂D_1. On ∂D_2 we have $\partial A_0 / \partial n = k_0$ (3.23).

Use the maximum principle for Laplace's equation. A_0 is a function of x and y, and is harmonic in the domain D. The boundary conditions are Neumann, *i.e.* $\mathbf{p}_0 \cdot \mathbf{n} = 0$ on ∂D_1, *i.e.* $\nabla A_0 \cdot \mathbf{n} = \partial A_0 / \partial n = 0$ on ∂D_1. The Hopf maximum principle says that the outward pointing normal derivative is strictly positive for non-constant functions. But A_0 does not have a strictly positive normal derivative (it is zero), so that A_0 must be constant. Hence we have that $\nabla A_0 = (p_0, q_0) = 0$, *i.e.* that at leading order, flux is zero.

Order ε

The momentum equation (3.19) at $\mathcal{O}(\varepsilon)$ is

$$h_0 \nabla h_1 + h_1 \nabla(h_0 + b) = 0,$$

but since $\nabla(h_0 + b) = 0$ and for $h_0 \neq 0$,

$$\nabla h_1 = 0.$$

Thus h_1 depends only on t. From the boundary condition (3.22) on ∂D_2, we have $h_1(x, y, t) = \phi_1(t)$ everywhere.

The mass equation (3.18) gives $\nabla \cdot \mathbf{p}_1 = -\phi_{1t}$. Defining $(p_1, q_1) = \nabla A_1$ for some function $A_1(x, y, t)$, we have

$$\nabla^2 A_1 = -\phi_{1t},$$
$$\text{with } \frac{\partial A_1}{\partial n} = 0 \quad \text{on } \partial D_1, \quad \text{and} \quad \frac{\partial A_1}{\partial n} = k_1(x, y) \quad \text{on } \partial D_2.$$

Since $\phi_1(t) \neq 0$, this equation is Poisson's equation with Neumann boundary conditions from (3.20) and (3.23). It can be solved by constructing a Green's function, $G(\mathbf{x}, \xi)$ that satisfies

i) $\nabla^2 G(\mathbf{x}, \xi) = -\delta(\mathbf{x} - \xi)$ for $\mathbf{x} \in D$,

ii) $\dfrac{\partial G}{\partial n} = 0$ on ∂D_1 and $\dfrac{\partial G}{\partial n} = 0$ on ∂D_2.

Using Green's second identity we obtain

$$A_1(\mathbf{x}) = \int_D G(\mathbf{x}, \xi) h_{1t}(\mathbf{x}, \xi)\, \mathrm{d}V + \int_{\partial D_2} G(\mathbf{x}, \xi) k_1(\mathbf{x})\, \mathrm{d}S,$$

where $k_1 = -b + \phi_1$ on ∂D_2 by (3.23). Therefore we can write

$$\mathbf{p}_1 = \nabla \left(\int_D G h_{1t}\, \mathrm{d}V + \int_{\partial D_2} G k_1\, \mathrm{d}S \right),$$

where $k_1 = -\mathbf{u}_1 b + \mathbf{u}_0 \phi_1$, but since $\mathbf{u}_0 = 0$, we have $k_1 = -\mathbf{u}_1 b$.

Order ε^2

The momentum equation (3.19) at $\mathcal{O}(\varepsilon^2)$ is

$$h_0 \nabla h_2 + h_1 \nabla h_1 + h_2 \nabla(h_0 + b) = 0.$$

But since $\nabla(h_0 + b) = 0$, and $\nabla h_1 = 0$, we are left with (provided $h_0 \neq 0$)

$$\nabla h_2 = 0.$$

Thus h_2 depends only on t. But the boundary condition (3.22) states $h_2 = 0$ on ∂D_2, *i.e.* for all t, so that $h_2(x, y, t) = 0$.

Now from the equation of mass (3.18) at $\mathcal{O}(\varepsilon^2)$,

$$\nabla \cdot \mathbf{p}_2 = 0.$$

Letting, as before, $\mathbf{p}_2 = \nabla A_2$ for some function $A_2(x, y, t)$, we have Laplace's equation with $\partial A_2/\partial n = 0$ on ∂D_1 from (3.20) and $\partial A_2/\partial n = k_2(x, y)$ on ∂D_2 from (3.23). Again, look for a Green's function G_2 that satisfies

i) $\nabla^2 G_2(\mathbf{x}, \xi) = -\delta(\mathbf{x} - \xi)$ for $\mathbf{x} \in D$,

ii) $\dfrac{\partial G_2}{\partial n} = 0$ on ∂D_1 and $\dfrac{\partial G_2}{\partial n} = 0$ on ∂D_2.

But since this Green's function satisfies the same conditions as G above, we let $G_2 = G$. Then Green's second identity gives

$$\begin{aligned} A_2(\mathbf{x}) &= \int_{\partial D_2} k_2(\mathbf{x}) G(\mathbf{x}, \xi)\, \mathrm{d}S, \\ \Rightarrow \quad \mathbf{p}_2 &= \nabla \left(\int_{\partial D_2} G k_2 \, \mathrm{d}S \right), \end{aligned}$$

where $k_2 = \mathbf{u}_1 \phi_1(t)$.

Order ε^3

The momentum equation (3.19) at $\mathcal{O}(\varepsilon^3)$ is

$$\nabla(h_0 h_3) + \nabla(h_1 h_2) + h_3 \nabla b = 0.$$

But since it was shown that $\nabla(h_0 + b) = 0$, that both ∇h_1 and ∇h_2 are zero, we are left with (provided $h_0 \neq 0$)

$$\nabla h_3 = 0.$$

Thus h_3 depends only on t. But the boundary condition (3.22) states $h_3 = 0$ on ∂D_2, *i.e.* for all t, so that $h_3(x, y, t) = 0$.

We find the same equation for the equation of mass (3.18) as at $\mathcal{O}(\varepsilon^2)$,

$$\nabla \cdot \mathbf{p}_3 = 0.$$

Following the same method, and letting $\mathbf{p}_3 = \nabla A_3$ for some function $A_3(x, y, t)$, we find

$$A_3(\mathbf{x}) = \int_{\partial D_2} k_3(\mathbf{x}) G(\mathbf{x}, \xi)\, \mathrm{d}S,$$
$$\Rightarrow \quad \mathbf{p}_3 = \nabla \left(\int_{\partial D_2} G k_3\, \mathrm{d}S \right),$$

where G is a Green's function that satisfies the properties stated above and where $k_3 = 0$, so that $A_3 = 0$, and hence $\mathbf{p}_3 = \mathbf{0}$.

3.2.3 Asymptotic solution

Each order of magnitude calculated adds to the accuracy of the solution, so at order ε^3 the solution is third order accurate. The solution is

$$h(x, y, t) = -b(x, y) + \varepsilon \phi_1(t) + \mathcal{O}(\varepsilon^4) \tag{3.24}$$

$$\mathbf{p} = (p, q) = \varepsilon \nabla \left(\int_D G h_{1t}\, \mathrm{d}V + \int_{\partial D_2} G k_1\, \mathrm{d}S \right) + \varepsilon^2 \nabla \int_{\partial D_2} G k_2\, \mathrm{d}S + \mathcal{O}(\varepsilon^4), \tag{3.25}$$

where $k_1 = -\mathbf{u}_1 b$, $k_2 = \mathbf{u}_1 \phi_1$, and G is the Green's function described above. A Green's function for a circular domain is constructed in Appendix C.1.

The solution indicates that the depth of the water is determined in the main part by the still water level and by the imposed boundary condition, *i.e.* by the force of the tide. Only at $\mathcal{O}(\varepsilon^4)$ or smaller do we see the effect of the advective terms on the water depth.

Figure 3.2 shows the surface elevation for the case of no source terms up to $\mathcal{O}(\varepsilon^3)$, with $b = -10$, $\phi_1(t) = \sin(2\pi t/T)$, where $T = 24$ hours.

3.2.4 Source terms

Equation (2.4) is the modification to the right hand side of the SWEs that models bed friction, Coriolis, and wind friction. Recall

$$\mathbf{s} = \begin{bmatrix} q_s \\ -gh\frac{\partial b}{\partial x} - gh\mathbf{S}_{fx} + fhv + c_w w^2 \tau_x \\ -gh\frac{\partial b}{\partial y} - gh\mathbf{S}_{fy} - fhu + c_w w^2 \tau_y \end{bmatrix}, \tag{3.26}$$

with

$$\mathbf{S}_f = \begin{bmatrix} \mathbf{S}_{fx} \\ \mathbf{S}_{fy} \end{bmatrix} = \begin{bmatrix} c_f u |\mathbf{u}| \\ c_f v |\mathbf{u}| \end{bmatrix}, \quad \tau = \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix} = \begin{bmatrix} \cos\alpha \\ \sin\alpha \end{bmatrix}.$$

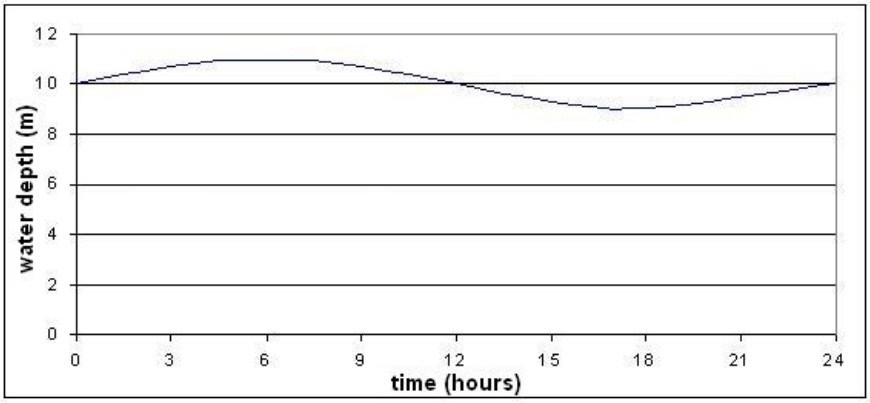


Figure 3.2: Order $\mathcal{O}(\varepsilon^3)$-accurate solution to the time evolution of surface elevation measured at an arbitrary point in the bay. The bathymetry is assumed flat ($b = -10$) and sinusoidal tidal effects ($\phi_1(t) = \sin(2\pi t/24)$, time in units of hours) are imposed at the boundary ∂D_2 from time $t = 0 \ldots 24$ hours.

In the mass equation, q_s represents the source/sink discharge per unit area (rainfall, evaporation), which is taken to be zero here. In the momentum equations, $\mathbf{S}_f$ represents the bed friction, with c_f a bed friction coefficient, f is the Coriolis parameter, and τ represents wind.

Now the mass equation is unchanged and the momentum equations become

$$\mathrm{Fr}^2\left(\frac{\partial(hu)}{\partial t}+\frac{\partial(hu^2)}{\partial x}+\frac{\partial(huv)}{\partial y}\right)+h\frac{\partial h}{\partial x}=-h\frac{\partial b}{\partial x}-B_f u|\mathbf{u}|+\frac{\mathrm{Fr}^2}{\mathrm{Ro}}hv+\mathcal{W}\cos\alpha$$
$$\mathrm{Fr}^2\left(\frac{\partial(hv)}{\partial t}+\frac{\partial(huv)}{\partial x}+\frac{\partial(hv^2)}{\partial y}\right)+h\frac{\partial h}{\partial y}=-h\frac{\partial b}{\partial y}-B_f v|\mathbf{u}|-\frac{\mathrm{Fr}^2}{\mathrm{Ro}}hu+\mathcal{W}\sin\alpha,$$

where Fr, the Froude number, Ro, the Rossby number, B_f and $\mathcal{W}$ are all dimensionless parameters which we discuss below.

Dimensions

Let us consider each source term in turn.

The bed friction coefficient, c_f, is written using Manning's n as $c_f = n^2 h^{-4/3}$. We take $n = \mathcal{O}(10^{-2})\,\mathrm{m}^{-1/3}\mathrm{s}$ in line with Chow (1959), Baines (1974) and Fowler (1997). The dimensionless bed friction parameter is

$$B_f = n^2 L D^{-7/3} U^2 = \mathcal{O}(10^{-10/3}) \approx \mathcal{O}(10^{-3}).$$

The Coriolis parameter represents the turning of the Earth, and is $f = 2\Omega \sin\varphi$, where Ω is angular velocity of the Earth, φ is the latitude of the domain. In the case of Moreton Bay, $\varphi = 27°\text{S}$ so $f = -6.6 \times 10^{-5} = \mathcal{O}(10^{-5})\,\text{s}^{-1}$, negative in the southern hemisphere. Here any variation in latitude is neglected: the β-plane approximation is taken, and f is assumed constant (Pedlosky (2003)). Furthermore, the nondimensional Rossby number,

$$\text{Ro} = \frac{U}{2\Omega \sin\varphi L} = \frac{U}{fL} = \mathcal{O}(10^{-1})$$

is small, so the Coriolis force is significant.

In the equations above, the size of the Coriolis term is therefore

$$\frac{\text{Fr}^2}{\text{Ro}} = \mathcal{O}(10^{-3}).$$

The wind term contains c_w the wind friction coefficient, w the wind velocity, and α the direction of the wind with respect to the x-axis. The wind friction coefficient can be taken to be $c_w = \mathcal{O}(10^{-3})$ (as done by Butman (1978), Trenberth *et al.* (1989), or Yu and O'Brien (1991)). We use Moreton Bay as a typical example: according to the Australian Government Bureau of Meteorology's website [1], the wind velocity may vary from 1.1 to $8.3\,\text{ms}^{-1}$, with an average of about $2.8\,\text{ms}^{-1}$. Thus we nondimensionalise with the dimensional value W such that $w = W\tilde{w}$ and $W = \mathcal{O}(1)\,\text{ms}^{-1}$. The valid range for α is $(0°, 360°)$.

The nondimensional wind term is now

$$\mathcal{W} = \frac{c_w W^2 L}{gD^2} = \mathcal{O}(10^{-1}).$$

Letting $\varepsilon = 10^{-1}$ as before, we write the system (3.17) with this particular choice (3.26) of source vector as

$$\frac{\partial h}{\partial t} + \frac{\partial (hu)}{\partial x} + \frac{\partial (hv)}{\partial y} = 0 \tag{3.28a}$$

$$\varepsilon^4 \left(\frac{\partial (hu)}{\partial t} + \frac{\partial (hu^2)}{\partial x} + \frac{\partial (huv)}{\partial y} \right) + h\frac{\partial h}{\partial x} = -h\frac{\partial b}{\partial x} - \varepsilon^3 h^{-1/3} u|\mathbf{u}| + \varepsilon^3 hv + \varepsilon \cos\alpha \tag{3.28b}$$

$$\varepsilon^4 \left(\frac{\partial (hv)}{\partial t} + \frac{\partial (huv)}{\partial x} + \frac{\partial (hv^2)}{\partial y} \right) + h\frac{\partial h}{\partial y} = -h\frac{\partial b}{\partial y} - \varepsilon^3 h^{-1/3} v|\mathbf{u}| - \varepsilon^3 hu + \varepsilon \sin\alpha. \tag{3.28c}$$

1. www.bom.gov.au/climate/averages/tables/cw_040842.shtml

Leading order

The system with source terms is identical to that without at leading order, so the solutions are

$$h_0 = -b(x, y)$$

$$\mathbf{p}_0 = \mathbf{0}.$$

Order ε

At order ε the momentum equations (3.28b) and (3.28c) in vector notation are

$$h_0 \nabla h_1 + h_1 \nabla (h_0 + b) = \tau,$$

where $\tau = [\cos\alpha, \sin\alpha]$. But $\nabla(h_0 + b) = 0$, thus

$$h_0 \nabla h_1 = \tau.$$

Take the divergence of both sides to obtain

$$\nabla^2 h_1 = \nabla \cdot \left(\frac{\tau}{h_0} \right),$$

and since the boundary conditions are $\partial h_1 / \partial n = 0$ on ∂D_1 (3.21) and $h_1 = \phi_1$ on ∂D_2 (3.22), means solving Poisson's equation with mixed boundary conditions, and h_1 can be expressed in terms of a Green's function H by defining

i) $\nabla^2 H(\mathbf{x}, \xi) = -\delta(\mathbf{x} - \xi)$ for $\mathbf{x} \in D$,

ii) $\dfrac{\partial H}{\partial n} = 0$ on ∂D_1 and $H = 0$ on ∂D_2.

Then by Green's second identity we obtain

$$h_1 = -\int_D H \nabla \cdot \left(\frac{\tau}{h_0} \right) \mathrm{d}V + \int_{\partial D_2} \phi_1 \frac{\partial H}{\partial n} \, \mathrm{d}S.$$

The mass equation (3.28a) at $\mathcal{O}(\varepsilon)$ is

$$\nabla \cdot \mathbf{p}_1 = -h_{1t}.$$

As before, define $\mathbf{p}_1 = \nabla A_1$ and construct a Green's function, G, so that we obtain

$$\mathbf{p}_1 = \nabla \left(\int_D G h_{1t} \, \mathrm{d}V + \int_{\partial D_2} G k_1 \, \mathrm{d}S \right),$$

where $k_1 = -\mathbf{u}_1 b$, and G is a Green's function.

Order ε^2

The momentum equation at $\mathcal{O}(\varepsilon^2)$ is

$$h_0\nabla h_2 + h_1\nabla h_1 + h_2\nabla(h_0+b) = 0.$$

But since $\nabla(h_0+b)=0$ and $\nabla h_1 = \tau/h_0$, we have

$$\nabla h_2 = -\frac{h_1\tau}{h_0^2}$$
$$\nabla^2 h_2 = \nabla\cdot\left(\frac{-h_1\tau}{h_0^2}\right).$$

And using Green's functions as above,

$$h_2 = \int_D H\nabla\cdot\left(\frac{h_1\tau}{h_0^2}\right)\mathrm{d}V,$$

where H is the Green's function with mixed boundary conditions. From the equation for mass (3.28a) at $\mathcal{O}(\varepsilon)$, we have

$$\mathbf{p}_2 = \nabla\left(\int_D Gh_{2t}\,\mathrm{d}V + \int_{\partial D_2} Gk_2\,\mathrm{d}S\right),$$

where G and H are Green's functions, and $k_2 = \mathbf{u}_1\phi_1$.

Order ε^3

At order ε^3, the momentum equations (3.28b) and (3.28c) are

$$h_0\nabla h_3 + h_1\nabla h_2 + h_2\nabla h_1 + h_3\nabla(h_0+b) = -h_0^{-1/3}\mathbf{u}_0|\mathbf{u}_0| + h_0\mathbf{k}\wedge\mathbf{u}_0,$$

but since $\nabla h_1 = \tau/h_0$, and $\nabla h_2 = -h_1\tau/h_0^2$, the equation reduces to

$$\nabla h_3 = -\frac{\tau}{h_0^3}\left(h_0h_2 - h_1^2\right).$$

Using Green's identities as before, both h_3 and $\mathbf{p}_3$ can be solved as

$$h_3 = \int_D H\nabla\cdot\psi_3\,\mathrm{d}V,$$
$$\mathbf{p}_3 = \nabla\left(\int_D Gh_{3t}\,\mathrm{d}V + \int_{\partial D_2} Gk_3\,\mathrm{d}S\right).$$

where G and H are Green's functions and

$$\psi_3 = -\frac{\tau}{h_0^3}\left(h_0h_2 - h_1^2\right), \quad k_3 = 0.$$

Putting this information together, we have the asymptotic solutions up to $\mathcal{O}(\varepsilon^3)$

$$h = -b(x,y) - \varepsilon\left(\int_D H\nabla\cdot\left(\frac{\tau}{h_0}\right)\mathrm{d}V - \int_{\partial D_2}\phi_1\frac{\partial H}{\partial n}\,\mathrm{d}S\right) + \varepsilon^2\int_D H\nabla\cdot\left(\frac{h_1\tau}{h_0^2}\right)\mathrm{d}V + \varepsilon^3\int_D H\nabla\cdot\psi_3\,\mathrm{d}V + \mathcal{O}(\varepsilon^4), \quad (3.29)$$

$$\mathbf{p} = (p,q) = \varepsilon\nabla\left(\int_D h_{1t}G\,\mathrm{d}V + \int_{\partial D_2}Gk_1\,\mathrm{d}S\right) + \varepsilon^2\left(\int_D h_{2t}G\,\mathrm{d}V + \int_{\partial D_2}Gk_2\,\mathrm{d}S\right) + \varepsilon^3\left(\int_D Gh_{3t}\,\mathrm{d}V\right) + \mathcal{O}(\varepsilon^4), \quad (3.30)$$

where G and H are Green's functions that satisfy Neumann and mixed boundary conditions, respectively and

$$k_1 = -\mathbf{u}_1 b, \quad k_2 = \mathbf{u}_1\phi_1, \quad \text{and} \quad \psi_3 = -\frac{\tau}{h_0^3}\left(h_0h_2 - h_1^2\right).$$

The effect of adding source terms of this size is evident at order ε and below. These effects are due to the wind alone: Coriolis and bed friction appear at $\mathcal{O}(\varepsilon^4)$ or below.

3.2.5 The importance of source terms

The inclusion of extra source terms in (3.28) results in corrections to the asymptotic solution to terms of order ε and below. We now use different values of firstly wind speed and secondly bed roughness in order to see their effects on the solutions. This provides insight into the behaviour of the source terms on the flow.

Effect of wind speed

Now consider the case when wind speed $W = 0.3\,\mathrm{ms}^{-1}$ so that $W^2 = \mathcal{O}(10^{-1})$ and $\mathcal{W} = \mathcal{O}(\varepsilon^2)$. The boundary conditions are unchanged.

The leading order and order ε momentum and mass equations are unchanged, thus

$$h_0 = -b, \quad h_1 = \phi_1$$
$$\mathbf{p}_0 = \mathbf{0}, \quad \mathbf{p}_1 = \nabla\left(\int_D Gh_{1t}\,\mathrm{d}V + \int_{\partial D_2}Gk_1\,\mathrm{d}S\right).$$

At order ε^2 the momentum equations (3.28b) and (3.28c) are

$$\begin{aligned} h_0\nabla h_2 + h_1\nabla h_1 + h_2\nabla(h_0+b) &= \tau \\ \Rightarrow \quad \nabla h_2 &= \frac{\tau}{h_0} \\ (\nabla\cdot) \quad \Rightarrow \quad \nabla^2 h_2 &= \nabla\cdot\left(\frac{\tau}{h_0}\right). \end{aligned}$$

Then look for a Green's function, H, and using the same method as before we have

$$h_2 = -\int_D H\nabla\cdot\left(\frac{\tau}{h_0}\right)\,\mathrm{d}V + \int_{\partial D_2} Hk_2\,\mathrm{d}S,$$

where $k_2 = \mathbf{u}_1\phi_1$. The order ε^2 equation for mass (3.28a) is

$$\nabla\cdot\mathbf{p}_2 = -h_{2t},$$

which leads to

$$\mathbf{p}_3 = \nabla\left(\int_D Gh_{2t}\,\mathrm{d}V + \int_{\partial D_2} Gk_2\,\mathrm{d}S\right).$$

At order ε^3, the momentum equation is

$$h_0\nabla h_3 + h_1\nabla h_2 + h_3\nabla(h_0+b) = -h_0^{-1/3}\mathbf{u}_0|\mathbf{u}_0| + h_0\mathbf{k}\wedge\mathbf{u}_0.$$

But since $\mathbf{u}_0 = \mathbf{0}$ we obtain $\nabla h_3 = -h_1\tau/h_0^2$. Taking the divergence of both sides and constructing a Green's function, H with mixed boundary conditions, we obtain

$$h_3 = \int_D H\nabla\cdot\psi_3\,\mathrm{d}V,$$

where H is a Green's function and

$$\psi_3 = -\frac{h_1\tau}{h_0^2}.$$

Since $h_{3t} \neq 0$, the equation of mass (3.28a) gives $\nabla\mathbf{p}_3 = -h_{3t}$ and by previous methods we have

$$\mathbf{p}_3 = \nabla\left(\int_D Gh_{3t}\,\mathrm{d}V\right),$$

where G is a Green's function.

This decrease in the wind term by one order leads to the following solution

$$h = -b + \varepsilon\phi_1 - \varepsilon^2\left(\int_D H\nabla\cdot\left(\frac{\tau}{h_0}\right)\,\mathrm{d}V - \int_{\partial D_2} Hk_2\,\mathrm{d}S\right) + \varepsilon^3\int_D H\nabla\cdot\psi_3\,\mathrm{d}V + \mathcal{O}(\varepsilon^4),$$

$$\mathbf{p} = (p,\, q) = \varepsilon \left(\int_D G h_{1t} \,\mathrm{d}V + \int_{\partial D_2} G k_1 \,\mathrm{d}S \right) + \varepsilon^2 \nabla \left(\int_D G h_{2t} \,\mathrm{d}V + \int_{\partial D_2} G k_2 \,\mathrm{d}S \right) + \varepsilon^3 \nabla \left(\int_D G h_{3t} \,\mathrm{d}V \right) + \mathcal{O}(\varepsilon^4)$$

where G and H are the usual Green's functions and

$$k_1 = -\mathbf{u}_1 b, \quad k_2 = \mathbf{u}_1 \phi_1, \quad \text{and} \quad \psi_3 = -\frac{h_1 \tau}{h_0^2}.$$

This shows that wind effects now appear at order ε^2, *i.e.* $\mathcal{O}(1)$ wind is responsible for effects of $\mathcal{O}(\varepsilon)$ and $\mathcal{O}(\varepsilon)$ wind is responsible for effects at $\mathcal{O}(\varepsilon^2)$.

Effect of bed roughness

This time, we assume $W = 1$ and the bed friction term is now $\mathcal{O}(\varepsilon^2)$, which corresponds to changing $n^2 = 10^{-4}$ to $n^2 = 10^{-3}$ (the bed becoming rougher). The equations are unchanged up to order ε^3, where the bed friction terms appear, but since $\mathbf{u}_0 = \mathbf{0}$, the equations reduce to those with source terms and hence the solution is unchanged from (3.29) and (3.30).

This shows that increasing the value of Manning's n from $n^2 = \mathcal{O}(\varepsilon^4)$ to $n^2 = \mathcal{O}(\varepsilon^3)$ results in corrections to the solution at $\mathcal{O}(\varepsilon^4)$ or smaller.

3.2.6 Asymptotic summary

These results are valid for the assumptions we have made, *i.e.* for the particular geometry, boundary conditions, and time scales of this bay, and are quite similar to the results of Vázquez-Cendón (1999), who performs asymptotic analysis for the shallow water equations on a similar geometry.

In Chapter 5, *Riemann2D* is run for different magnitudes of the source terms for some idealised bays. These tests allow us to verify, using the above knowledge, that *Riemann2D* is able to produce realistic and physically relevant results.

3.2.7 Stratified shallow water

The shallow water equations are an approximate model that tell us information about the global behaviour of the system. We may choose to refine our model, as we did by adding source terms,

or alternatively by considering that the water consists of two layers of unmixed, interacting fluids.

Density-stratified, unmixed fluids can be found in places where one fluid flows into another, as in the case of the Strait of Gibraltar where the (saltier, denser) Mediterranean Sea flows under the (less salty and less dense) Atlantic Sea (as has been studied by Lane-Serff and Woodward (2001), Farmer and Armi (1988)). An alternative motivation is to model the different forces acting on different parts of the water, so that the fluid at the surface is subject to the force of the wind more than to the roughness of the bed, and vice versa for fluid near the bed.

Dimensional and asymptotic analysis of the stratified flow problem can be found in Appendix C.2. We summarise the results here. Let the variables in the upper fluid be $(h,\, p,\, q)$ with density ρ_1, and in the lower fluid $(H,\, P,\, Q)$ and the density ρ_2. The density ratio is defined to be $\rho_1/\rho_2 = \gamma$. The depth of still water in the upper layer is $-b_*$, a constant, and in the lower layer $-b+b_* = -b(x,y)+b_*$. Then the asymptotic solution, based on the nondimensionalisation of (C.5) and with boundary conditions analogous to those of the previous section, is

$$
\begin{aligned}
h &= -b_* + \varepsilon^2\phi_2(t) + \mathcal{O}(\varepsilon^4) \\
H &= -b + b_* + \varepsilon\phi_1(t) + \mathcal{O}(\varepsilon^4) \\
\mathbf{p} &= \varepsilon\nabla\int_{\partial D_3} Gk_1 \,\mathrm{d}S + \varepsilon^2\nabla\left(\int_D Gh_{2t}\,\mathrm{d}V + \int_{\partial D_3} Gk_2\,\mathrm{d}S\right) + \mathcal{O}(\varepsilon^4) \\
\mathbf{P} &= \varepsilon\nabla\int_D GH_{1t}\,\mathrm{d}V + \varepsilon^2\nabla\int_{\partial D_2} GK_2\,\mathrm{d}S + \mathcal{O}(\varepsilon^4).
\end{aligned}
$$

It can be seen how similar these solutions are to the single-layer solutions, as we would expect from the similarity of the boundary conditions. A more physically representative solution may be obtained by adjusting the relative importance of the bed friction and wind terms for each layer.

It would be interesting to develop the numerical aspect of this problem and evaluate the benefits of such a model, but it is beyond the scope of this thesis.

3.3 Summary

In this chapter we aimed to show the capabilities and the limitations of analysis for the shallow water equations. The main points of this chapter have been

- *Introduction of some classical ideas about hyperbolic systems and the shallow water equations, calculation of eigenvalues and Riemann invariants for the two-dimensional system, and definition of weak solutions. In particular, the eigenvalues and the idea of weak solutions are central to the design of a numerical scheme.*

- *Nondimensionalisation of the shallow water equations with source terms to show the relative importance of each term. Presentation of asymptotic solutions, in the case of no source terms and non-varying bathymetry to show the dependence on bathymetry and tide of the water depth.*

- *Variation of the size of the wind speed and bed friction terms. Decreasing wind speed from* $w^2 = \mathcal{O}(1)$ *to* $w^2 = \mathcal{O}(\varepsilon)$ *reduced wind effects from* $\mathcal{O}(\varepsilon)$ *to* $\mathcal{O}(\varepsilon^2)$*. Increasing the size of the bed friction term so that* $n^2 = \mathcal{O}(\varepsilon^4)$ *changed to* $n^2 = \mathcal{O}(\varepsilon^3)$ *had no noticeable effects above* $\mathcal{O}(\varepsilon^4)$*.*

- *Asymptotic analysis of the stratified system of SWEs. This approach may lead to more physically realistic results as it treats the top layer as being subject to different forces from the bottom.*

The most important part is the asymptotic study of the source terms, as this will be used to compare with numerical results from *Riemann2D* in later chapters.

The next chapter looks at the design of a numerical scheme for a hyperbolic problem to understand how *Riemann2D* is built.

CHAPTER 4

Numerical schemes

Riemann2D is a shock-capturing, object-oriented numerical solver designed for the shallow water equations in two spatial dimensions, which uses an FVM and unstructured triangular meshes. It was written in Java by Jha (2006) and others, originally for short-time problems such as the dam break problem. This thesis is concerned with using the *Riemann2D* project on much larger (both spatial and temporal) problems, where flows are generally slower.

In this chapter a clear explanation is given of how a PDE or a system of PDEs is formulated into an FVM scheme, which is based on standard textbook explanations. The use of limiters to ensure that spurious oscillations do not arise and to prevent solutions being numerically 'diffused' is discussed. An outline of the main properties of object-oriented languages (such as Java) is given, and we discuss in some detail the structure of the *Riemann2D* packages.

A newly-implemented time-dependent boundary condition to reproduce the behaviour of a tide has been added to the shallow water package and is described in this chapter. Finally, it is shown how the advection equation that is used to model contaminant transport is used in *Riemann2D* .

4.1 Numerical schemes

A numerical scheme provides an approximate solution to a single equation or a system of equations with the associated boundary conditions. It is a powerful way of solving equations

under flow conditions that are extremely unlikely to yield explicit analytical solutions.

In this section we show how a general one- or multi-dimensional hyperbolic system is formulated into a finite volume scheme. We also consider requirements on the scheme that ensure accuracy and some ways to ensure a scheme achieves this via the addition of a *limiter*. This chapter provides an overview of the FVM and its implementation in *Riemann2D* .

4.1.1 Finite volume formulation: 1D

Consider a general homogeneous 1D hyperbolic system of PDEs such as

$$\mathbf{q}_t + f(\mathbf{q})_x = 0. \tag{4.1}$$

The problem is formulated into a finite volume method in the following way, as described by LeVeque (2003). Consider a space divided into cells or elements Ω_i of length $\Delta x = x_{i+1/2} - x_{i-1/2}$, as shown in Figure 4.1. For simplicity the cells are assumed equal in length. Define an average flux across a cell as

$$\mathbf{Q}_i^n \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{q}(x, t_n)\, \mathrm{d}x. \tag{4.2}$$

This is the finite volume that gives the method its name. Then the integral form of (4.1) is

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_i} \mathbf{q}(x,t)\, \mathrm{d}x = F(\mathbf{q}(x_{i-1/2}, t)) - F(\mathbf{q}(x_{i+1/2}, t)).$$

This is the semi-discrete algorithm: to obtain the fully explicit time-marching algorithm, integrate over one timestep from t_n to t_{n+1}

$$\int_{\Omega_i} \mathbf{q}(x, t_{n+1})\, \mathrm{d}x - \int_{\Omega_i} \mathbf{q}(x, t_n)\, \mathrm{d}x = \int_{t_n}^{t_{n+1}} F(\mathbf{q}(x_{i-1/2}, t)) - F(\mathbf{q}(x_{i+1/2}, t))\, \mathrm{d}t.$$

Rearranging, dividing by Δx, and making use of (4.2) we see that (4.1) can be written

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{\Delta x} \left(F_{i+1/2}^n - F_{i-1/2}^n \right),$$

where the intercell flux $F_{i-1/2}$ is a function of its left and right states, *i.e.* $F_{i-1/2} = \mathcal{F}(Q_{i-1}, Q_i)$ for some function $\mathcal{F}$ to be defined.

This is the fully explicit time-marching finite volume formulation that can be used to provide an approximate numerical solution. The choice of the flux function $\mathcal{F}$ is discussed in the following section (recall also Section 2.2.5).

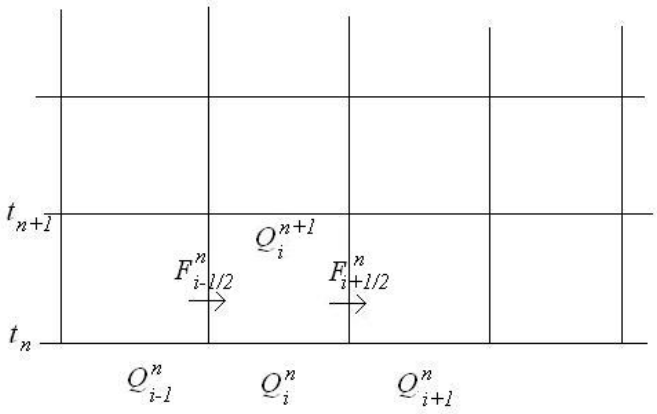


Figure 4.1: One-dimensional grid on which the problem may be solved. Reproduced from LeVeque (2003). Note that although the arrows of the fluxes are depicted pointing to the right, the flux is not necessarily in that direction.

Roe's scheme

A possible choice of the intercell flux function is

$$F_{i-1/2} = \frac{1}{2}\left(F_{i-1}^n + F_i^n\right) - \frac{1}{2}|\mathbf{R}_{i-1/2}^n|\left(\mathbf{q}_i^n - \mathbf{q}_{i-1}^n\right),$$

where $\mathbf{R}$ is some matrix to be defined. This is the form that Roe's solver uses.

Roe (1981) had the seminal idea to solve an approximate problem in which the flux vector did not depend on the varying solution, but on a (locally constant) approximation to it.

Roe's idea was to replace the Jacobian matrix of variables with a matrix of constants that are local approximations to the variables. These approximations are averages of the left and right states of the data. The new matrix must also satisfy some properties – collectively known as property U – in order for the scheme to be *uniformly* valid across discontinuities (conservative, hyperbolic), as set out in Section 2.2.4.

For the 1D shallow water equations in the form

$$\mathbf{q}_t + F(\mathbf{q})_x = 0, \qquad \Leftrightarrow \qquad \mathbf{q}_t + A\mathbf{q}_x = 0,$$

where $A = \partial F/\partial \mathbf{q}$ is the Jacobian matrix of the flux vector F. Then the Roe averages can easily be shown to be (see for example Glaister (1988a))

$$\sqrt{\tilde{h}} = \sqrt{h_L} + \sqrt{h_R} \tag{4.3}$$

$$\tilde{u} = \frac{(hu)_L/\sqrt{h_L} + (hu)_R/\sqrt{h_R}}{\sqrt{\tilde{h}}}. \tag{4.4}$$

This means that the matrix (of variables) A, can be replaced with the matrix (of constants) $\tilde{A}$, where

$$\tilde{A} = \begin{bmatrix} 0 & 1 \\ g\tilde{h} - \tilde{u}^2 & 2\tilde{u} \end{bmatrix}.$$

Thus for a 1D nonlinear hyperbolic scheme the flux function is written as

$$F_{i-1/2} = \frac{1}{2}\left[\tilde{A}\mathbf{Q}_{i-1} + \tilde{A}\mathbf{Q}_i - \sum_{p=1}^{m} |\tilde{\lambda}_p| a_{i-1/2,p}\tilde{\mathbf{r}}_p\right], \tag{4.5}$$

where $\tilde{\lambda}_p$ is the pth eigenvalue of $\tilde{A}$, $a_{i-1/2}$ is the wavestrength measured at $i-1/2$, and $\tilde{\mathbf{r}}_p$ is the pth eigenvector corresponding to the pth eigenvalue of $\tilde{A}$.

The wavestrength is determined by the definition (Roe (1981))

$$\Delta\mathbf{Q} = \mathbf{Q}_i - \mathbf{Q}_{i-1} = \sum_{p=1}^{m} a_{i-1/2,p}\mathbf{r}_p. \tag{4.6}$$

Hence for the shallow water equations, the wavestrengths are given by

$$a_{i-1/2} = \begin{bmatrix} \left((\tilde{u}+\tilde{c})(h^R_{i-1/2} - h^L_{i-1/2}) - ((hu)^R_{i-1/2} - (hu)^L_{i-1/2})\right)/2\tilde{c} \\ \left(-(\tilde{u}-\tilde{c})(h^R_{i-1/2} - h^L_{i-1/2}) + ((hu)^R_{i-1/2} - (hu)^L_{i-1/2})\right)/2\tilde{c} \end{bmatrix},$$

where $\tilde{c} = \sqrt{g(eta - bedElevation)}$, $eta = \frac{1}{2}(h^R_{i-1/2} + h^L_{i-1/2})$, $bedElevation = \frac{1}{2}(b^R_{i-1/2} + b^L_{i-1/2})$, and the tildes denote the Roe average (visualisation of left and right state values is given in Figure 4.2).

Finally we obtain Roe's method for the 1D shallow water equations

$$\mathbf{Q}^{n+1}_i = \mathbf{Q}^n_i - \frac{\Delta t}{2\Delta x}\left(\tilde{A}(\mathbf{Q}^n_{i+1} + \mathbf{Q}^n_{i-1}) + \sum_{p=1}^{m} |\tilde{\lambda}^n_p|(a^n_{i+1/2,p} - a^n_{i-1/2,p})\tilde{\mathbf{r}}^n_p\right). \tag{4.7}$$

The first term of the right hand side is known for each element, the terms in brackets are calculated for each element, then the left hand side is updated accordingly.

4.1.2 Multidimensional finite volume formulation

We wish to extend the FVM to higher dimensions. Consider a two-dimensional scheme of the form

$$\mathbf{q}_t + F(\mathbf{q})_x + G(\mathbf{q})_y = 0 \tag{4.8}$$

such as the shallow water equations. Consider also a Cartesian grid in the (x, y) plane. The x-axis is divided into N blocks, and for convenience we define $\Delta x = x_{i+1/2} - x_{i-1/2}$. Similarly, the y-axis is divided into M blocks, and we define the length $\Delta y = y_{j+1/2} - y_{j-1/2}$. The grid is time-stepped over in steps of size Δt.

Consider the homogeneous, hyperbolic, two-dimensional system (4.8). We wish to arrange the system into a scheme resembling Roe's 1D scheme (4.7), which we do following Namin *et al.* (2004) or Brufau and Garcia-Navarro (2000).

The usual one-dimensional flux average for FVM is

$$\mathbf{Q}_i^n \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{q}(x, t_n)\, \mathrm{d}x.$$

If working with a structured (rectangular) mesh, this is extended to a two-dimensional control volume in a natural way

$$\mathbf{Q}_{i,j}^n \approx \frac{1}{\Delta x \Delta y} \int_{y_{i-1/2}}^{y_{i+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{q}(x, y, t_n)\, \mathrm{d}x\, \mathrm{d}y.$$

However, a different formulation is required for the unstructured triangular meshes used here. To see clearly where each term comes from, start from (4.8), and consider a cell of arbitrary volume Ω, over which $\mathbf{q}$ is conserved. Integrate over this volume to obtain

$$\begin{aligned} &\iint_\Omega \mathbf{q}_t + \nabla \cdot \mathbf{F}\, \mathrm{d}x\, \mathrm{d}y = 0 \\ \Rightarrow \quad &\frac{\partial}{\partial t} \iint_\Omega \mathbf{q}\, \mathrm{d}x\, \mathrm{d}y + \iint_\Omega \nabla \cdot \mathbf{F}\, \mathrm{d}x\, \mathrm{d}y = 0, \end{aligned}$$

where $\mathbf{F} = (F, G)$ the vector of flux vectors. If $\mathbf{q}$ is sufficiently smooth, we may use the divergence theorem to obtain

$$\frac{\partial}{\partial t} \iint_\Omega \mathbf{q}\, \mathrm{d}x\, \mathrm{d}y + \int_{\partial\Omega} \mathbf{n} \cdot \mathbf{F}\, \mathrm{d}s = 0,$$

where s is an arclength parametrisation of $\partial\Omega$, $\mathbf{n} = (n_x,\, n_y)$ is the unit outward-pointing vector out of $\partial\Omega$, and now $\mathbf{F} = \mathbf{F}(s)$. Now integrate from t_n to t_{n+1} and divide by the cell area, $|\Omega|$, so

$$\frac{1}{|\Omega|}\iint_\Omega \mathbf{q}(x,y,t_{n+1})\,\mathrm{d}x\,\mathrm{d}y - \frac{1}{|\Omega|}\iint_\Omega \mathbf{q}(x,y,t_n)\,\mathrm{d}x\,\mathrm{d}y + \frac{1}{|\Omega|}\int_{t_n}^{t_{n+1}}\int_{\partial\Omega}\mathbf{n}\cdot\mathbf{F}\,\mathrm{d}s\,\mathrm{d}t = 0. \quad (4.9)$$

As LeVeque (2003) points out, in the 1D formulation, we are looking for a scheme of the form

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2}^n - F_{i-1/2}^n\right).$$

This suggests we re-write (4.9) by defining

$$\mathbf{Q}_{ij}^n \approx \frac{1}{|\Omega|}\iint_\Omega \mathbf{q}(x,y,t_n)\,\mathrm{d}x\,\mathrm{d}y,$$

the 2D approximation to $\mathbf{q}$ over volume Ω, and

$$\breve{\mathbf{F}}_k^n \approx \frac{1}{\Delta t}\int_{t_n}^{t^{n+1}}\left(\frac{1}{l_k}\int_{\mathrm{side}_k}\mathbf{n}\cdot\mathbf{F}\right)\mathrm{d}t, \quad (4.10)$$

the fluxes through each side of the cell Ω. Here l_k is the length of side k. This allows us to write

$$\mathbf{Q}_{ij}^{n+1} = \mathbf{Q}_{ij}^n - \frac{\Delta t}{|\Omega|}\sum_{k=1}^{N} l_k\breve{\mathbf{F}}_k^n, \quad (4.11)$$

which is the 2D equivalent to the 1D Roe solver (4.7), and for a triangular mesh, $N = 3$. (The above applies to a system of dimension p by extending in the obvious way.) The first term of the right hand side is known, the second is calculated for each cell and the left hand side is updated accordingly for each timestep. The detail of the implementation of this in *Riemann2D* is given in later sections.

The expression $\breve{\mathbf{F}}_k^n$ is determined by using the expression for the intercell flux function (4.5) in the integral (4.10) for each of the sides of the triangular element.

The advantage of the FVM formulation is that it can be applied to a mesh composed of any type of polygon, and in any dimension. The elements do not need to be arranged in a structured way, and therefore is an ideal choice for a problem on an arbitrary domain.

Roe's solver

Roe's solver in 2D is exactly analogous to the 1D solver: replacing the flux matrices of variables F and G with approximate matrices of constants $\tilde{F}$ and $\tilde{G}$, defined via the Roe averages for the state variables on the left and right sides of each side of the element. Thus we find the eigenvalues, eigenvectors and wave speeds of the approximate matrix

$$n_x\tilde{F} + n_y\tilde{G} = \begin{bmatrix} 0 & n_x & n_y \\ (g\tilde{h} - \tilde{u}^2)n_x - \tilde{u}\tilde{v}n_y & 2\tilde{u}n_x + \tilde{v}n_y & \tilde{u}n_y \\ (g\tilde{h} - \tilde{v}^2)n_y - \tilde{u}\tilde{v}n_x & \tilde{v}n_x & \tilde{u}n_x + 2\tilde{v}n_y \end{bmatrix},$$

where tilde denotes the usual Roe averages

$$\sqrt{\tilde{h}} = \sqrt{h_L} + \sqrt{h_R}$$
$$\tilde{u} = \frac{(hu)_L/\sqrt{h_L} + (hu)_R/\sqrt{h_R}}{\sqrt{\tilde{h}}} \quad \tilde{v} = \frac{(hv)_L/\sqrt{h_L} + (hv)_R/\sqrt{h_R}}{\sqrt{\tilde{h}}},$$

as given in Namin *et al.* (2004) and Brufau and Garcia-Navarro (2000), for example.

4.1.3 Accuracy and stability

A successful numerical scheme should approximate the solution without smearing sharp regions or introducing spurious oscillations. It should approximate the solution increasingly accurately as the number of grid cells is increased, with the property that the numerical solution tends to the actual solution as the grid size tends to zero. These requirements can be formalised by stating that a numerical scheme should have the following properties:

Consistency The discrete operator (on finite differences) converges towards the continuous operator (on partial derivatives) as the grid is refined, *i.e.* as Δt, $\Delta x \to 0$.

Stability 'Noise' from initial conditions or other, does not grow.

Convergence The numerical solution converges towards the real solution as Δt, $\Delta x \to 0$.

For a finite difference scheme, we use Lax's equivalence theorem, which states that for linear problems, a necessary and sufficient condition for convergence is that the method is both consistent and stable (see for example Versteeg and Malalasekera (1995)). This is a useful theorem, since consistency and stability are easier to prove than convergence. However, since we are using finite volumes, we must look more closely at the above conditions.

Stability criteria

Since the solution to a hyperbolic problem is a series of interacting waves, we must ensure that the computational grid is such that the time step Δt is small enough that the wave does not have time to exit the other side of the element of width Δx (Font (2003)). This leads us to the following important condition.

The Courant-Friedrichs-Lewy (Courant *et al.* (1928)) condition (or CFL condition) states that the numerical domain of dependence must contain the physical domain of dependence.

Definition The number $|c|\Delta t/\Delta x$ is known as the the Courant number, where c is the wavespeed, which for the shallow water equations is given by $c = \sqrt{gh}$. It is a necessary condition for the stability of a scheme. Thus satisfying

$$|c|\frac{\Delta t}{\Delta x} < 1$$

is a requirement for a scheme to be successful.

For the two-dimensional case, Jha (2006) has derived a formula that generalises the CFL criterion for triangular meshes. It uses the wavespeed, c, the length of one of the sides of the triangular element, l, and the area of the triangle A, or that of an adjacent triangle A_i:

$$\frac{\Delta t}{2}\max_{i=1,2,3}\left(\frac{c_i l_i}{\min(A, A_i)}\right) \leqslant 1.$$

Details of how this is used in the present numerical scheme is given later in this chapter.

Numerical order of accuracy

Consider the scalar advection equation $q_t + Fq_x = 0$ for some scalar F. Let the operator $\mathcal{H}_k$ be a method such as Lax-Friedrichs or Beam-Warming, so for example the Lax-Friedrichs $\mathcal{H}_k$ is (LeVeque (1990))

$$\mathcal{H}_k(Q^n; j) = \frac{1}{2}(Q^n_{j-1} + Q^n_{j+1}) - \frac{\Delta t}{2\Delta x}(Q^n_{j+1} - Q^n_{j-1}),$$

where $k = \Delta t$ the size of the timestep, and Q^n_j is the approximate solution to q at time $t = n$ and point $x = j$.

Definition The *truncation error* is defined to be

$$L_k(x,t) = \frac{1}{\Delta t}\left[q(x,t+\Delta t) - \mathcal{H}_k(q(\cdot,t);x)\right],$$

where $k = \Delta t$ is the size of the timestep.

Thus the truncation error for the Lax-Friedrichs scheme is

$$L_k(x,t) = \frac{1}{\Delta t}\left[q(x,t+\Delta t) - \frac{1}{2}\left(q(x-\Delta x,t) + q(x+\Delta x,t)\right)\right] + \frac{1}{2\Delta x}F\left(q(x+\Delta x,t) - q(x-\Delta x,t)\right).$$

Taylor expanding q this becomes

$$\begin{aligned} L_k(x,t) &= q_t + Fq_x + \frac{1}{2}\left(\Delta t q_{tt} - \frac{\Delta x^2}{\Delta t}q_{xx}\right) + \mathcal{O}(\Delta x^2) \\ &= \frac{1}{2}\Delta t\left(F^2 - \frac{\Delta x^2}{\Delta t^2}I\right)q_{xx} + \mathcal{O}(\Delta x^2). \end{aligned}$$

Note that as $\Delta x \to 0$, the truncation error tends to $\mathcal{O}(\Delta t)$ so the scheme is first-order accurate.

In general we may say that a scheme has pth order of accuracy if the truncation error, L_k, can be written

$$|L_k| \leqslant C\Delta x^p$$

for some constant C, and where p is the largest such number. See for example LeVeque (1990).

It is known (Jha (2006)) that Roe's scheme is first-order accurate, but is made high-resolution by the addition of a limiter.

The $\mathcal{C}$-property

Bermúdez and Vázquez (1994) give a conservation property that the scheme must satisfy if it is to conserve the quantities of the state variables.

Definition A numerical scheme is said to satisfy the $\mathcal{C}$-property if it solves exactly the steady-state problem.

Definition A numerical scheme is said to satisfy the *approximate* $\mathcal{C}$-property if it solves the steady-state problem to an accuracy of $(\Delta x)^2$, where Δx is the size of the computational grid.

Roe's scheme for SWEs with no bed friction is known to satisfy the exact $\mathcal{C}$-property (Vázquez-Cendón (1999)). This property will be demonstrated for *Riemann2D* in Chapter 5.

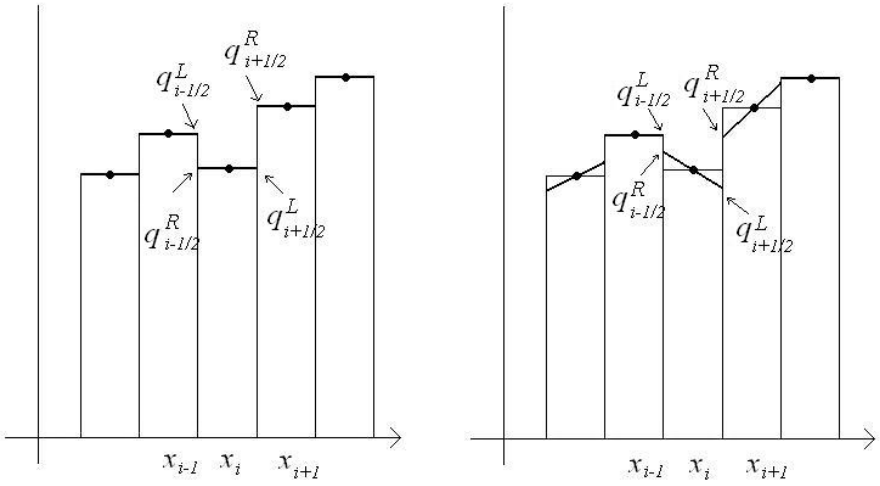


Figure 4.2: Comparison of the data reconstruction techniques of Godunov (left) with MUSCL (right). The Godunov method approximates the data in each cell with a piecewise *constant* value. MUSCL-type schemes use a *linear* approximation. Left and right states (q^L and q^R, respectively) are also shown.

4.1.4 Data reconstruction

A Godunov-type scheme takes the state values in each cell and replaces them with piecewise constant values (Hirsch (1984)), thus defining a series of local Riemann problems, and a first-order accurate reconstruction.

An extension to Godunov-type schemes that are used in this project are MUSCL-type schemes (Monotonic Upstream-centred Schemes for Conservation Laws, due to van Leer (Hirsch (1984), p494)). This means that the data are replaced with piecewise *linear* values. The two different approaches are represented in Figure 4.2. For a 2D (triangular mesh) problem, a plane must be fitted to replace the data, based on the state values in the three neighbouring triangles (*i.e.* triangle 123 in Figure 4.3).

The piecewise linear values of the MUSCL approach must be determined by some method. In this project, we use Roe's method, as described earlier. Other methods that might be used

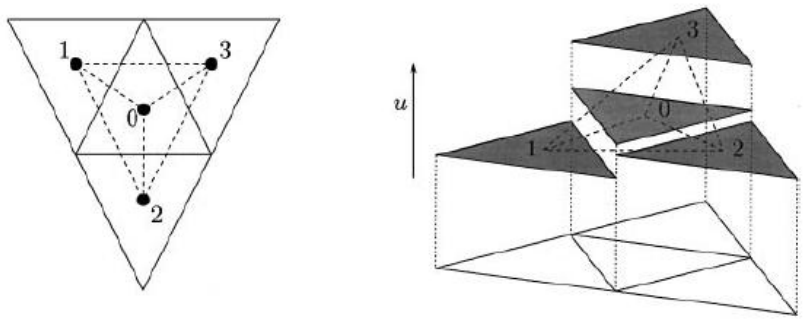


Figure 4.3: Naming convention for the limiting procedure as defined for a triangular control volume (left) and a piecewise constant reconstruction of the solution (right). Reproduced from Hubbard (1999).

include those of Harten, Lax, and van Leer, (the HLL and HLLC schemes, 'C' standing for 'contact'), and Shu-Osher. These have been used by Jha (2006) in his thesis in the context of *Riemann2D* .

Limiters

Although it is second-order accurate, the MUSCL approach can fail to accurately capture shocks or contact discontinuities when the gradient of the slope is too large, as this can introduce undershoots or overshoots in the numerical solution. The remedy for this is to *limit* the gradient of the slope by the addition of a nonlinear term, so that the solution is limited or not, depending how steep the gradient of the slope is. This idea of limiters was proposed by Boris and Book (1973) and van Leer (Hirsch (1984), p127).

Consider a triangular element of centre 0. Its three adjacent triangular elements have their centres at 1, 2, and 3 (see Figure 4.3). The gradient of the slope of the triangle $\Delta 123$ will be limited if it is too steep, *i.e.* the state variable will be re-written as

$$\mathbf{q}' = \mathbf{q} + \Phi(\mathbf{r}, \mathbf{L}),$$

where Φ is the chosen limiter, $\mathbf{r}$ is a vector from the centre of the triangle, and $\mathbf{L}$ is the gradient of $\Delta 123$. Some limiters available for use with *Riemann2D* are described below.

Superbee/minmod This limiter is due to Roe. It uses the gradient of triangle $\Delta 123$, and is defined as

$$\Phi_{\text{superbee/minmod}} = \min(\Phi_j),$$

where

$$\Phi_j = \min\left\{\max[\min(\beta r_j, 1), \min(r_j, \beta)]\right\}$$

where $1 \leqslant \beta \leqslant 2$ and for each component of the vector $\mathbf{r}$, r_j, we have

$$r_j = \begin{cases} (q_0^{\max} - q_0)/(q_j - q_0) & \text{if} \quad q_j - q_0 > 0 \\ (q_0^{\min} - q_0)/(q_j - q_0) & \text{if} \quad q_j - q_0 < 0 \\ 1 & \text{if} \quad q_j - q_0 = 0 \end{cases}$$

where

$$q_0^{\min} = \min(q_0, q_{\text{neighbour}}), \quad q_0^{\max} = \max(q_0, q_{\text{neighbour}}),$$

and where the notation $u_{\text{neighbour}}$ indicates the state variable in elements 1, 2 or 3. The case $\beta = 1$ corresponds to the minmod limiter, $\beta = 2$ to the superbee limiter.

Extended van Leer Let the gradients S_1, S_2, S_3 represent the gradients of triangles $\Delta 103$, $\Delta 203$ and $\Delta 302$ respectively. Then a limiter based on van Leer's limiter of 1974 is defined as

$$\Phi_{\text{EVL}} = 0.8 \frac{S_1 \cdot |S_2| \cdot |S_3| + |S_1| \cdot S_2 \cdot |S_3| + |S_1| \cdot |S_2| \cdot S_3}{|S_1| \cdot |S_2| + |S_2| \cdot |S_3| + |S_1| \cdot |S_3|}.$$

MLG This stands for Maximum Limited Gradient, and was developed by Batten *et al.* (1996). It has been shown to reduce to the superbee limiter of Roe in one dimension.

First calculate the gradient of each of the four triangles in Figure 4.3, so that

$$\Phi_0 = \Phi_{\text{LCD}}(\Delta 123), \qquad \Phi_1 = \Phi_{\text{LCD}}(\Delta 120)$$

$$\Phi_2 = \Phi_{\text{LCD}}(\Delta 103), \qquad \Phi_3 = \Phi_{\text{LCD}}(\Delta 023),$$

where the function Φ_{LCD} is the Limited Central Difference limiter so that $\Phi_{\text{LCD}} = (\min \alpha_j)\nabla(\Delta 123)$, where α_j is a scalar and ∇ is the unlimited gradient operator (see Batten *et al.* (1996) and Jha (2006)). Then set $\Phi_{\text{MLG}} = \Phi_i$ such that

$$|\Phi_i| = \max_{0 \leqslant j \leqslant 3} |\Phi_j|.$$

Sweby (1984) notes that in the tests he performs on van Leer's limiter, the minmod and superbee limiters of Roe, and Chakravarthy and Osher's limiter, the superbee limiter is very accurate, though has a tendency to over-compress, meaning initially smooth data can become squarish. Meanwhile, the limiter of van Leer "exhibits results nearly as good as Roe's whilst being more reliable" (Sweby (1984)).

The idea of the *rank* of a limiter was introduced by Jha (2006). The rank of a limiter depends on the number of triangles whose gradients are used by that limiter. Thus a scheme that uses no limiter is said to have a zero rank limiter, rank one limiters include the superbee and minmod limiters of Roe, and the limited central difference (LCD) limiter. A rank three limiter is for example the Extended van Leer, and the maximum limited gradient (MLG) limiter is rank four. There is an inverse correlation between the rank of a limiter and the amount of numerical diffusion, thus we expect the MLG limiter to best preserve the true solution.

4.1.5 Mesh generation

Control volumes in this project all have information centred at the element, rather than at the nodes. This means the control volume is the element itself: if a node-centred program is used, a volume must be created around each node, which is more computationally expensive and thus undesirable. The advantage in using an element-centred program is that it contains more computational points than a node-centred program. Thus it can be said that the element-centred program gives more information (Namin *et al.* (2004)).

A node-centred scheme more correctly conserves mass. This is because flux through the elements' sides is exactly calculated with a node-centred scheme, but is based on a linear plane approximation in an element-centred scheme. Thus the amount of fluid passing from one cell to the next is automatically exact with a node-centred scheme. However, since in this project we are dealing with slow flows, we may expect that any loss due to this type of averaging will be negligible.

The size of the elements in the mesh determines the timestep that should be used, if the Courant number is to be respected. If the elements are large, the timestep may also be large, but not so much so that information entering the cell at one side may pass to the other side before the timestep is updated.

It is possible to use a structured grid as the basis of the calculations (*e.g.* Borthwick *et al.* (2001b), Borthwick *et al.* (2001a), Geller (2003)). A structured mesh has the advantage of being easily generated – no specialised software is required. On the other hand, it is not well-suited to irregular boundaries, so an unstructured mesh is a more suitable choice.

Argus ONE® meshmaker was used to generate all the meshes in this thesis.

4.2 Object Orientation

The term 'object oriented programming' (OOP) was first used by Alan Kay in around 1967 (Kay (2003)) when the concept was first being developed. Today, Kay says that "OOP to me means only messaging, local retention and protection and hiding of state-processes, and extreme late binding of all things". In other words, the state of an object is kept locally in the object, not elsewhere.

An object-oriented (OO) language like Java or C++ allows or encourages the use of OO principles. The characteristic components of an object oriented program (also OOP) are **classes**, **methods** and **objects**. A class is a collection of methods and fields (methods and fields can be thought of as functions and variables, respectively). An object is an instance of a class: it is the concrete where a class is the theoretical or abstract.

There are at least four important programming techniques associated with OOP that make it particularly powerful.

Inheritance In OOP, a class can be 'subclassed', or made more specialised. [1] The subclass inherits all the properties – fields and methods – of the superclass from which it is created. Crucially, it is able to override the superclass to make its methods more relevant.

Encapsulation Although often confused with information hiding, encapsulation is in fact a way of enabling information hiding. Encapsulating is bundling the data with the methods operating on those data. In this way it is possible to shield client classes from the internal workings of a class. [2] In procedural languages, encapsulation is not possible. OOP on the other hand allows data and methods to be grouped so that data and the methods that

1. 'Subclass' is the syntax of Java and Smalltalk. In C++ the term 'derived class' is used.
2. `www.javaworld.com/javaworld/jw-05-2001/jw-0518-encapsulation.html`

act on them are in the same class. Information hiding is a design principle that prevents data from being changed by client classes, which might lead to spurious results. It also means that the superclass may change its private data without affecting the way a client accesses it.

Abstraction A powerful aspect of OOP is the use of abstract classes, which means that the class is independent of concrete implementations. An instance of an (abstract) class is a (concrete) object.

Polymorphism This refers to the ability of one method to act on different types of argument. For example, a method `CalculatePay()` might take as its argument either `salariedEmployee`, or `hourlyPaidEmployee`. Although both types of employee have data such as name, bank details, *etc.* , the method `CalculatePay()` operates in different ways: for `salariedEmployee`, the method gets the annual salary, divides by 12, and returns the answer; for `hourlyPaidEmployee`, the method gets the number of hours worked and the hourly pay, multiplies the two, and returns the result.

The two main advantages of using object-oriented programming is that it allows complex code to be written, and more importantly, this code is easily extensible. In the case of the *Riemann2D* project, there exists a `Generic` package that contains all the properties that are necessary to solve a general hyperbolic PDE. The `ShallowWater` package is specific to the shallow water equations, and inherits all the properties of the `Generic` package. It would be possible to write a package called `GasDynamics`, for example (see Figure 4.4), that solves Euler's equations for gas dynamics (another hyperbolic set of PDEs), and makes no reference to `ShallowWater`, but does inherit all the properties of `Generic`.

Historically, the most common language for mathematical problems to be solved in has been FORTRAN. This language is suited to such tasks as it matches the mathematician's approach in carrying out one step at a time. Object orientation has tried to replace this relatively narrow approach to programming, which only allows one problem to be solved by one code, by giving objects properties and then making the objects work together. The danger is that information can become hidden, and in a complex code may become overlooked, especially by a part of a programming team unaware of its presence. Thus a change made by one part of the team

may cause unwanted effects or apparently inexplicable errors. It is therefore most important that there is communication between team members. One way this can be facilitated is via a repository, where team members submit their code to once they are happy with a change they have made. Other team members then access the new updated version and can implement the changes so that each member of the team is always using the latest version.

4.2.1 The importance of Java

The power of Java is in its portability: Java is a 'write once, run anywhere' language, because it is written and compiled on one machine, with the code complied to an intermediate language, Java bytecode. When a user wants to run this code on their own computer, the bytecode is run on a Virtual Machine (VM). This means that Java is a 'safe' language, with aspects like memory allocation being taken care of away from the control of the programmer. It is also a strongly typed language, meaning anything in the code must be a *type*, *e.g.* int, double, string and so on. This has two implications – it prevents a programmer from writing code that may contain a memory leak,[1] but it also removes a certain amount of power from the programmer in writing code. Java is a high level language, so it looks more like English than machine code. This makes it quicker to learn and more intuitive to use than a lower-level language.

4.3 Object oriented *Riemann2D*

In the rest of this chapter, variables and expressions used in a mathematical or analytical context are written in the traditional way: h, u, v, η and so on. Those referring to anything done by *Riemann2D* are written in `typewriter text` to make them easily distinguishable.

4.3.1 Overview of *Riemann2D*

The structure of the *Riemann2D* project is represented in Figures 4.4 and 4.5. At the top in Figure 4.4 is the package containing the superclasses, `Generic`. At the next level are the `ShallowWater` and `Limiters` packages, and the (hypothetical) `GasDynamics` package. These packages inherit everything from `Generic`, and override or add as required. `Limiters` may be

1. A memory leak can occur when memory is allocated to a program then not freed when it is no longer needed.

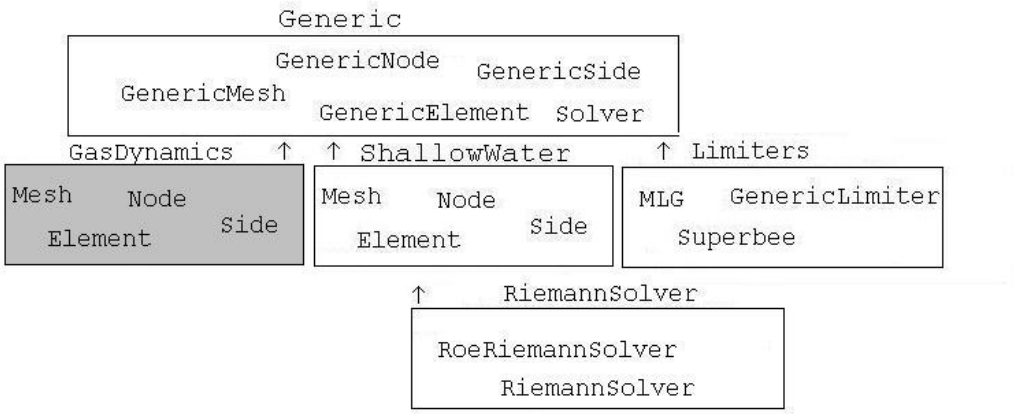


Figure 4.4: Representation of the structure of *Riemann2D* . The top package, `Generic`, is the superclass from which other subclasses inherit. The `ShallowWater` package contains classes needed for solving the shallow water equations, and inherits the generic properties of element, node *etc.* from `Generic`, adding to them or overriding them as required. The `Limiters` package inherits properties from `Generic`. It is not linked to `ShallowWater` in the sense that it can be used with another package, *e.g.* `GasDynamics`, but it is available for `ShallowWater` to use. `RiemannSolver` contains subclasses of `ShallowWater` because it uses the eigenvalues and eigenvectors of the shallow water equations in its calculations.

called by `ShallowWater`, or by a class from another package that has the `Generic` classes as its superclass, *e.g.* `GasDynamics` in Figure 4.4. The `RiemannSolver` package is specific to the `ShallowWater` package, and cannot be used by another, *e.g.* `GasDynamics`.

The *Riemann2D* project requires a domain to be input in the form of a meshfile. The domain is first divided into a triangular grid using specialised software, then exported as a meshfile. The average flux over each of the elements of the grid is then calculated, which is (the equivalent to) the control volume of (4.2). The user/programmer specifies a meshfile for the program to read. An example of an input meshfile for a 2D domain is shown in Figure 4.6. The information is arranged in a way that allows parameter values to be read by the program. Each element has state variables [1] associated with it, *i.e.* $\mathbf{q} = [h, hu\, hv, (hc)]^{\mathrm{T}}$, as well as a

1. For the basic shallow water equations there are three state variables, for shallow water with a concentration

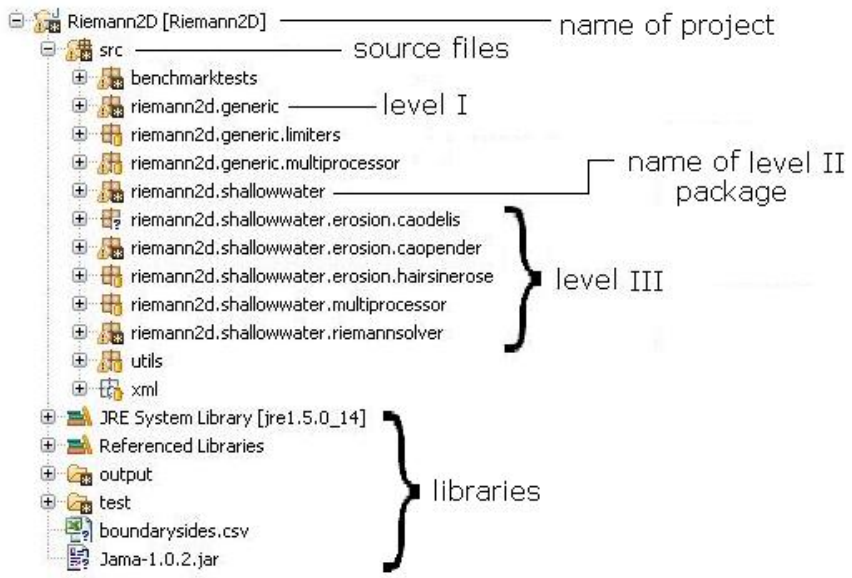


Figure 4.5: Alternative representation of the structure of *Riemann2D* . The package `Generic` is a level one package, containing all the superclasses. The `ShallowWater` package is a level two package, and contains classes needed for solving the shallow water equations; it inherits the generic properties of element, node *etc.* from `Generic`, adding to them or overriding them as required. The `Limiters` package is also level two, and inherits properties from `Generic`. `RiemannSolver` is level three and contains the subclass of `ShallowWater` because it uses the eigenvalues and eigenvectors of the shallow water equations in its calculations.

parameter which is usually b, bed elevation. The first line contains the information [number of nodes] [number of node parameters] [number of elements] [number of state variables] [number of element parameters]. The next two lines contain information about the size of the domain. The lines beginning `N` describe the node number and the coordinates of the node. The lines beginning `E` describe the element number, the three node numbers that belong to that element, and the depth of that element measured from some fixed datum level. The lines beginning `S` describe the boundary conditions. More explanation on boundary conditions is given later in this chapter.

A meshfile such as the one in Figure 4.6 is small and may be generated by the user/programmer

equation with one class of concentration there are four.

```
5       1       4       3       0
-1      0.25    1
-1      0.25    1
N       1       0       0       -1
N       2       -1      -1      -1
N       3       1       -1      -1
N       4       -1      1       -1
N       5       1       1       -1
E       1       1       2       3       0       0
E       2       1       4       2       0       0
E       3       1       3       5       1       0
E       4       5       4       1       1       0
S       1       S       1       0       0
```

Figure 4.6: Typical 2D node-centred input meshfile. The first line reads "no. of nodes, no. of node parameters, no. of elements, no. of state variables, no. of element parameters". Lines two and three give information about the dimensions of the domain and the number of divisions there. Lines beginning `N` contain "node number, x-coordinate, y-coordinate, `bedElevation`". Lines beginning `E` contain "element number, the three nodes that make up the triangle, `eta`, u, v". Finally, lines beginning `S` represent boundary sides: the side number, the boundary condition type, `eta`, u, v. For boundary sides that are left unspecified, the default condition is applied.

'by hand' by choosing suitable numbers, but for larger meshes we use the software Argus ONE® meshmaker, which automatically numbers the elements and nodes correctly. When the mesh is read in, *Riemann2D* then links elements together by checking which elements are adjacent. If an element is on the boundary of the domain, one or two of its neighbouring elements may be *ghost* elements. Ghost elements are elements created for convenience, and lie outside of the domain. At the interface of every pair of elements the Riemann problem is solved, thus a ghost element allows the problem to be solved on the boundary of the domain.

Once the mesh is constructed, a new instance of the class `Solver` is called. `Solver` calls `solve_outer`, and within this, `solve_inner` is called, which in turn calls `mesh.execute()`. For each timestep, the `mesh.execute` method is called. This is described in the diagram in Figure 4.7.

Within the `execute` loop, `leftState` and `rightState` define the left and right states of each element (see Figure 4.2) and calculate the slope of the bed elevation (or ∇b), respectively. The method `applySource()` provides modifications to the state variables from calculations based

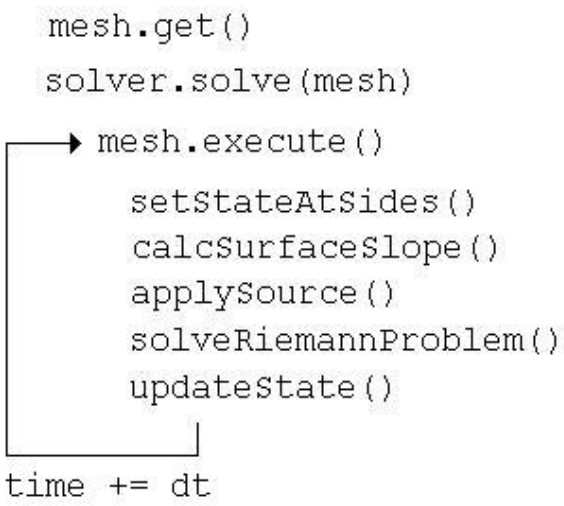


Figure 4.7: Methods that are called when *Riemann2D* is run with the shallow water model.

on the right hand side of (3.28) in the previous chapter.

`solveRiemannProblem()` allows the user/programmer to choose which solver to use. In this work we always choose Roe's solver. The Riemann problem is solved at each timestep on each of the three sides of every element for a particular choice of solver, then the maximum wavespeed is calculated and the Courant number is updated based on this wavespeed.

4.3.2 More detail on *Riemann2D*

The notation used in this section is as follows. The dependent variables of the problem are (h, hu, hv) = (`state[0]`, `state[1]`, `state[2]`). (The vector $\mathbf{q}$ is expressed in *Riemann2D* as `state`; the ith element of the vector $\mathbf{q}$ is chosen by writing `state[i]`.) The independent variables are `x`, `y`, `t`. Triangular control volumes (elements) are traversed in an anticlockwise manner with sides numbered as in Figure 4.8.

The primitive variables are initialised locally from the state variables as `h=state[0]-bedElevation`, `u=state[1]/h` and `v=state[2]/h` for non-zero `h`. Then the source terms are applied based on

the following equations:

```
source[0] = excessRainfallRate
```

```
source[1] = coriolis_factor*v - kinematicBedStress[0] +
            windStress_factor*cos(windDirection) - g*h*surfaceSlope[0]
```

```
source[2] = -coriolis_factor*u - kinematicBedStress[1] +
            windStress_factor*sin(windDirection) - g*h*surfaceSlope[1],
```

where `coriolis_factor` is f, so the first term is Coriolis force, the second

$$\texttt{kinematicBedStress[0]=n*n*g*h}^{-1/3}\texttt{*}\sqrt{\texttt{u*u+v*v}}\texttt{*u},$$

$$\texttt{kinematicBedStress[1]=n*n*g*h}^{-1/3}\texttt{*}\sqrt{\texttt{u*u+v*v}}\texttt{*v},$$

is force due to bed friction (`n` is Manning's n), and the third

```
windStress_factor = windStressCoefficient*windSpeed*windSpeed*rhoAir/rhoWater
```

is the force due to wind speed. The term `surfaceSlope` in `source[1]` and `source[2]` represents the b derivative, and is calculated by fitting a slope to the side of the element.

The method `roeApproximateRiemannSolver()` is where the mathematical concepts of Section 4.1 are used. We look at each stage of this method and compare it with Section 4.1.

If h_L, h_R – the left and right water depths, respectively – are smaller than the bed elevation, the program resets them to be $b_{L,R}$+`minDepth`, where `minDepth` is small and specified by the programmer. This prevents problems with vanishing water depth and friction terms becoming too large. The Roe averages used by *Riemann2D* are (compare with (4.3))

$$\sqrt{\texttt{h}_\texttt{R}} = \sqrt{\eta_\texttt{R} - \texttt{b}_\texttt{R}}, \quad \sqrt{\texttt{h}_\texttt{L}} = \sqrt{\eta_\texttt{L} - \texttt{b}_\texttt{L}},$$

$$\sqrt{\tilde{\texttt{h}}} = \sqrt{\texttt{h}_\texttt{R}} + \sqrt{\texttt{h}_\texttt{L}},$$

$$\tilde{\texttt{u}} = \frac{(\texttt{hu})_\texttt{L}/\sqrt{\texttt{h}_\texttt{L}} + (\texttt{hu})_\texttt{R}/\sqrt{\texttt{h}_\texttt{R}}}{\sqrt{\tilde{\texttt{h}}}}, \quad \tilde{\texttt{v}} = \frac{(\texttt{hv})_\texttt{L}/\sqrt{\texttt{h}_\texttt{L}} + (\texttt{hv})_\texttt{R}/\sqrt{\texttt{h}_\texttt{R}}}{\sqrt{\tilde{\texttt{h}}}}.$$

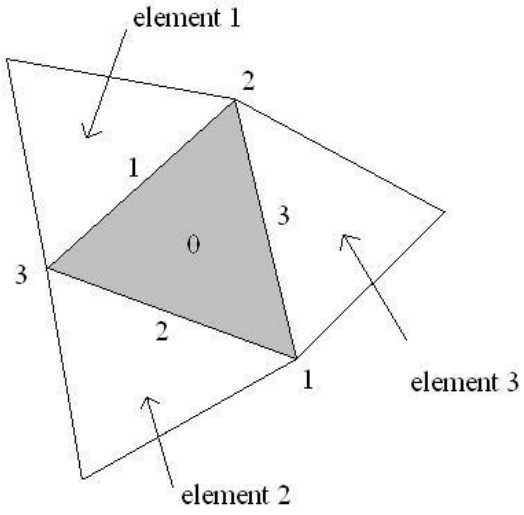


Figure 4.8: Example element, showing numbering of nodes and sides. Note that side i is opposite node i, and the sides and nodes are numbered in an anti-clockwise manner, starting from one. (Reproduced from Jha (2006).)

Jumps from left to right states are defined as

$$\Delta h = h_R - h_L, \quad \Delta hu = (hu)_R - (hu)_L, \quad \Delta hv = (hv)_R - (hv)_L,$$

$$\Delta hun = \Delta hu^* n_x + \Delta hv^* n_y, \quad \Delta hvn = -\Delta hv^* n_x + \Delta hu^* n_y$$

$$un = u^* n_x + v^* n_y, \quad vn = -v^* n_x + u^* ny.$$

Furthermore, define

$$eta = (h_R + h_L)/2, \quad bedElevation = (b_R + b_L)/2, \quad c = \sqrt{g(eta - bedElevation)}.$$

The wavespeeds, as a function of the jumps, are

$$a[i] = \begin{bmatrix} a[0] \\ a[1] \\ a[2] \end{bmatrix} = \begin{bmatrix} ((\tilde{u} + c)\Delta h - \Delta hun) \\ vn^* \Delta hvn \\ (-(\tilde{u} - c)\Delta h + \Delta hun) \end{bmatrix}.$$

Then the eigenvectors and eigenvalues are defined by the program as

$$wave[i][j] = \begin{bmatrix} 1 & 0 & 1 \\ \tilde{u} - c^* n_x & -n_y & \tilde{u} + c^* n_x \\ \tilde{v} - c^* n_y & n_x & \tilde{v} + c^* n_y \end{bmatrix}. \quad s[i] = \begin{bmatrix} s[0] \\ s[1] \\ s[2] \end{bmatrix} = \begin{bmatrix} |un - c| \\ |un| \\ |un + c| \end{bmatrix}.$$

Thus the term corresponding to the Q-scheme term $(\mathbf{Q}_i - \mathbf{Q}_{i+1})$ (4.6) is defined

$$\texttt{fluxDifference[i]} = \begin{bmatrix} \texttt{a[0]s[0]wave[0][0]} + \texttt{a[1]s[1]wave[0][1]} + \texttt{a[2]s[2]wave[0][2]} \\ \texttt{a[0]s[0]wave[1][0]} + \texttt{a[1]s[1]wave[1][1]} + \texttt{a[2]s[2]wave[1][2]} \\ \texttt{a[0]s[0]wave[1][0]} + \texttt{a[1]s[1]wave[1][1]} + \texttt{a[2]s[2]wave[1][2]} \end{bmatrix}.$$

The method `calcNumericalFluxFunction()` is called, which performs

$$\texttt{flux[i]} = (\texttt{rightFlux[i]} + \texttt{leftFlux[i]} - \texttt{fluxDifference[i]})/2^{*}\texttt{length}.$$

This is equivalent to the 2D Roe flux of (4.10)

$$\breve{\mathbf{F}}_k^n \approx \frac{1}{\Delta t}\int_{t_n}^{t^{n+1}} \left(\frac{1}{l_k}\int_{\text{side}_k} \mathbf{n}\cdot\mathbf{F}\right) \mathrm{d}t.$$

This is the outline of the functions of `roeApproximateRiemannSolver()`. The next computational step returns to the `mesh.execute()` method, where it remains to update the Courant number and the state values. For each element of the mesh, two important calculations are made

$$\begin{aligned} \texttt{sumFlux} &= \sum_{\texttt{j=0}}^{2} \texttt{unitFactor[j]}^{*}\texttt{side[j].flux[i]}, \\ \texttt{state0[i]} &= \texttt{state[i]} + \texttt{dt}^{*}(\texttt{source} - \texttt{sumFlux/area}). \end{aligned} \tag{4.12}$$

It is equivalent to the right hand side of (3.28) in Chapter 3. It can clearly be seen that `sumFlux` is equivalent to $\sum_k l_k \breve{\mathbf{F}}_k$, and that (4.11)

$$\mathbf{Q}_{ij}^{n+1} = \mathbf{Q}_{ij}^n - \frac{\Delta t}{|\Omega|}\sum_{k=1}^{N} l_k \breve{\mathbf{F}}_k^n$$

is equivalent to (4.12). This method is performed for every pair of elements of the mesh, and repeated for every timestep.

4.3.3 Boundary conditions

Boundary conditions can be the same for all sides of the boundary, in which case the type of condition may be specified in the program as default type (reflective or transmissive types). If boundary conditions are different in different parts of the boundary domain, or if special values are required, these must be specified in the meshfile.

There are four original types of boundary condition in *Riemann2D* : state type (S), transmissive type (T), reflective type (R) and flux type (F). The reflective type is a subset of flux

type and the transmissive type is a subset of state type. The first pair (R, F) specify a value (flux) for the state variables at the side. For R, this value is zero, for F it is some user-specified value. No Riemann problem is solved, since the value at the side is known. The values used for the flux type are specified in the meshfile, and thus is not available as a default type condition. Reflective type on the other hand, may be used as default.

The other two types (T, S) require a Riemann problem to be solved. Transmissive type sets the state variables of the ghost cell to be equal to those of the interior cell, and solves the Riemann problem across the interface. It is "a numerical attempt to produce boundaries that allow the passage of waves without any reflection" (Jha (2006)), and sets $\mathbf{Q}_{\text{ghost element}} = \mathbf{Q}_{\text{boundary element}}$. State type sets the state variables of the ghost cell to be equal to some user-specified values, and solves across the interface. The transmissive type is available as a default condition.

The four types are summarised in Table 4.1.

F	S
state at side = values	state in ghost = values
✗ Riemann problem	✓ Riemann problem
R	**T**
state at side = 0	state in ghost = state in interior
✗ Riemann problem	✓ Riemann problem

Table 4.1: The four original types of boundary condition for this project.

New implementation of time-dependent boundary condition in *Riemann2D*

A time-dependent condition has been added to *Riemann2D* to simulate the tidal forcing felt by a natural bay, equivalent to condition (3.22). In this 'periodic' type (P) of boundary condition, `state[0]` has a state type boundary condition that is updated at each timestep, and the other state variables have reflective type. For the boundary condition on `state[0]` the ghost cell is

$$\texttt{element}[0].\texttt{state}[0] = \texttt{amp} * \texttt{Math.cos}(\texttt{freq} * (\texttt{Solver.time} - \texttt{phase})),$$

where the coefficients should be chosen to represent the physical reality. This condition is applied every timestep and a Riemann problem is solved across the boundary. The other state variables are updated according to the transmissive boundary condition.

This boundary condition is applied to some idealised bays in Chapter 6, and to a real bay in Chapter 7.

4.3.4 Contaminants

Contaminant transport is modelled in *Riemann2D* with a simple advection equation, following Benkhaldoun *et al.* (2007), Periáñez (2004), and Lin and Falconer (1997) (neglecting diffusion). For each extra class of contaminant, the following equation is solved

$$(hc_i)_t + (huc_i)_x + (hvc_i)_y = 0,$$

where c_i represents the concentration of the ith class. In this thesis we consider only one class of contaminant, which we denote c. Where there is ambiguity, we denote wavespeed $c = \sqrt{gh}$.

This equation is added to the system and solved by *Riemann2D* in the following way:

- increase the number of columns for the elements in the meshfile and increase the number of state variables by one;
- assign initial values of concentration to the required elements;
- *Riemann2D* solves the equation in the same way as the equation for mass, with the unknown variable `state[3]`$= hc$ rather than `state[0]`$= \eta$.

The boundary condition for `state[3]` should be chosen so as to be physically relevant. For example, for a continuous release of a contaminant, the boundary condition would need to be set at each timestep in the same way as for `state[0]`. Chapters 6 and 7 make use of this equation to trace the path of a pollutant in a bay.

This thesis proposes a generic model for pollutant transport – we are not considering a specific pollutant. Thus we used a nominal value of $0.3\,\mathrm{kg\ m^{-3}}$. Other values that have been used are: $0.1\,\mathrm{mg\ l^{-1}}$ (Periáñez (2004)), 0.05-3.78 $\mathrm{mg\ l^{-1}}$ (Kachiashvili *et al.* (2007)), 100 units (Lin and Falconer (1997)), and 10 units (Benkhaldoun *et al.* (2007)). The first two papers had measurements of particular substances that caused harm in the bay of interest, and the second

two used values that demonstrated the effectiveness of their scheme, so the values were nominal. This provides ample justification for our choice of $0.3\,\mathrm{kg\ m^{-3}}$, as we are not measuring a specific pollutant, we merely aim to demonstrate the general fate of the pollutant.

4.3.5 Wetting and drying cells

When a domain has sloping sides at a boundary, where the slope is shallow and therefore subject to flooding then drying, this provides a numerical challenge. When the water depth goes to zero, as we have already seen, the shallow water equations loose their hyperbolic property, but even if h remains nonzero but gets sufficiently small, *Riemann2D* can encounter difficulties when dividing by this small number, leading to numerical instability and unphysical results.

To fix this problem, a value of the minDepth parameter is set in the program, so that whenever a cell has a water depth less than this, it is set to be the value of minDepth. This effectively covers the dry cells with a small layer of water, that is nevertheless large enough to prevent numerical instabilities developing.

A recent version of *Riemann2D* follows the same approach as Liang *et al.* (2006) and Namin *et al.* (2004) (Chapter 2) in 'freezing' the flux out of an element found to have zero depth, then re-introducing it once the depth has increased. This development has been too recent to include any results from it here, but it will be a valuable function to be able to model flooding and drying, especially when the applications are to real bays where in practice this occurs.

4.4 Summary

The most important things covered in this chapter are

- *Demonstration of the formulation of a general hyperbolic PDE for the FVM in one and two dimensions. In two dimensions, the divergence rule is applied, meaning the scheme is suitable for calculations on domains of arbitrary geometry as studied here.*

- *Description of data reconstruction using the MUSCL approach, and how the order of accuracy can be improved by the addition of a limiter. Some of the limiters that form part of Riemann2D were presented. The advantage of using an unstructured mesh is in*

its easy application to irregular boundaries, and an element-centred mesh provides more information.

- *The power of object orientation is its extensibility. Some other of its useful properties such as encapsulation, inheritance and polymorphism were also described.*
- *Detail of the structure of Riemann2D – the generic superclasses can be used to solve any type of hyperbolic equation, while the shallow water subclasses contain information specific to that system, including source terms, eigenvalues and wavespeeds.*
- *Implementation of a time-dependent boundary condition to Riemann2D that can be used to mimic the tidal forcing of the ocean.*
- *Description of the treatment of a passive pollutant, using a purely advective equation.*

This chapter gives a clear idea about how *Riemann2D* works and details the advantages of using an OO language. In the next chapter we test the code on some simple problems to check that it is performing well and giving realistic results.

CHAPTER 5

Testing *Riemann2D*

This chapter forms the validation stage of the code testing in which we determine the degree to which the simulation is an accurate model of the real world. In the first part we use two different test domains and vary the size of the bed friction and wind terms in order to measure the importance of these parameters to the flow. Results are compared with the asymptotic analysis of Chapter 3. In the second part, we use two analytical solutions to the 2D SWEs to demonstrate the benefit of using a limiter and to determine the accuracy of *Riemann2D* .

The two domains used in this chapter are:

- a square of constant depth – this is the simplest 2D domain and that which most closely matches the assumptions of the asymptotic solution, and in which we expect the flow to be quasi-1D and influences of bed friction to be negligible because of the relatively large and uniform h
- a circular basin – this is simple in that it is axisymmetric, but has vanishingly small depths at the edge which allows the problems associated with modelling bed friction to be verified. It is also the domain on which we find the analytical solutions that enable us to demonstrate the code's accuracy.

All tests in this chapter are for subcritical flows with typical velocity of around $0.1\,\mathrm{ms}^{-1}$ or less and long time (three hours or more); tests for high-speed flows and short time using *Riemann2D* have been made by Jha (2006).

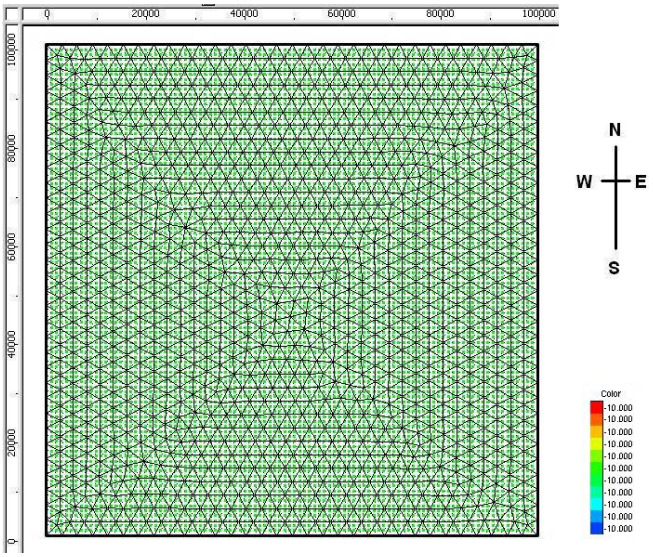


Figure 5.1: Mesh of 2326 elements of a square domain of 100 km square and uniform depth of 10 m.

The software Argus ONE® meshmaker is used to generate all of the test meshes of this thesis. One of the square meshes used in this chapter is shown in Figure 5.1. The mesh is exported from Argus to a meshfile, which can then be read by *Riemann2D*.

5.1 Square domain

The first series of tests were carried out on a square domain with a side length of 100 km and a constant depth of 10 m. Three mesh densities with ratio approximately 10 were used: 234 elements, 2326 elements (shown in Figure 5.1) and 23192 elements.

5.1.1 Effect of different wind speeds

The effect of different wind speeds was tested by simulating wind blowing across the surface of the bay in a westerly direction for three hours at three different wind speeds. Although the wind direction was westerly, results should be invariant whether wind is northerly, southerly or easterly. The fine mesh (23192 elements) was used, with a fixed timestep of dt = 1 s, the minimum depth parameter, minDepth $= 1 \times 10^{-6}$ m, the wind friction coefficient 1×10^{-3} and

the three wind speeds were 20, 2 and 0.2 ms^{-1}. The Met Office website [1] gives details of the Beaufort wind force scale: 20 ms^{-1} is classified as a fresh gale (scale 8), 2 ms^{-1} is a light breeze (scale 2) and 0.2 ms^{-1} is calm conditions (scale 0), so these values represent a realistic range, as well as differing by an order of magnitude one from the next. The Manning's n and Coriolis parameter were zero to allow only wind effects to be seen. The tests were stopped after a simulation time of three hours. The time taken to run the simulations was around 60 minutes for each of the three runs.

Recall the expression for energy from (3.13),

$$E = \int_A \frac{1}{2}\rho h|\mathbf{u}|^2 + \frac{1}{2}\rho g h^2 \,\mathrm{d}A, \tag{5.1}$$

where A is the area of the domain. From this it is calculated that the total initial, steady-state (potential) energy of this system is 4.905×10^{15} J, or $E_0 = 4.905 \times 10^5$ J per unit area, which is in exact agreement with *Riemann2D*. Table 5.1 shows the energy added to the system per unit area (composed of kinetic energy ΔKE and potential energy ΔPE) calculated by *Riemann2D* for each of the tested wind speeds, from which we see that the energy added is much smaller compared to E_0. Both kinetic and potential energy vary with wind speed and we see that $\Delta KE \ll \Delta PE$ because of the small effect of bed friction.

In Chapter 3 it was shown that a change in wind speed of one order leads to a change in the solution of two orders. Table 5.1 shows that $\tilde{\mathbf{p}}$ varies by two orders for a change in wind speed of one order, thus agreeing with the analysis.

Effect of refining the mesh

The above test was repeated with the same wind speeds using the medium and the coarse meshes. Table 5.2 gives information about the Courant numbers for these runs.

It can be seen that the Courant number increases by a factor of about three for an increase in mesh density of one order. This is expected, for, as the CFL condition states, as $\Delta x \to 0$, the timestep must also be refined to prevent the Courant number going above one and numerical instabilities developing. Thus as the mesh is refined, the Courant number increases if other factors remain unchanged.

1. The Beaufort scale can be found on the Met Office website at
`www.metoffice.gov.uk/weather/marine/guide/beaufortscale.html`

wind speed	$\Delta E = \Delta KE + \Delta PE$ (J = kg m^{2}s^{-2})	$\Delta E/E_0$	$\tilde{\mathbf{p}} = \text{avg}(h\sqrt{u^2+v^2})$ (m^{2}s^{-1})
$w = 20$	$4.730 \times 10 + 3.565 \times 10^3$	7.364×10^{-3}	8.426×10^{-1}
$w = 2$	$4.822 \times 10^{-3} + 3.544 \times 10^{-1}$	7.323×10^{-7}	9.300×10^{-3}
$w = 0.2$	$4.822 \times 10^{-7} + 3.544 \times 10^{-5}$	7.323×10^{-11}	9.586×10^{-5}

Table 5.1: Energies and velocity calculated by *Riemann2D* for different values of wind speed w at time $t = 3$ hours. The first column shows added calculated energy per unit area, made of kinetic energy + potential energy. The second shows energy added as a proportion of original energy. The third column shows the average fluxes.

	Courant number		
wind speed	234	2326	23192
$w = 20$	0.0017	0.0054	0.0169
$w = 2$	0.0016	0.0050	0.0160
$w = 0.2$	0.0016	0.0050	0.0160

Table 5.2: Courant number for three wind speeds $w = 20$, $w = 2$ and $w = 0.2\,\text{ms}^{-1}$, on meshes of three different densities of 234, 2326, and 23192 elements.

We see that the Courant number is small and therefore satisfies the CFL condition. This is the case for all numerical tests in this thesis, and is necessary to ensure stability.

Demonstration of the C-property

The test was run again with zero initial velocity and no forcing at the boundary. The wind speed, Manning's n and Coriolis parameters were all zero, and the solution was updated at each timestep. As there is no force acting on the domain, we expect the initial velocity and surface elevation to be maintained at the end of the three-hour simulation. Indeed, if the scheme does maintain this, it is said to satisfy the C-property. It was found that the zero solution after three hours was exactly maintained, which agrees with Vázquez-Cendón (1999).

Manning's n	$\Delta E = \Delta KE + \Delta PE$ (J = kg m^{2}s^{-2})	$\Delta E/E_0$	$\tilde{\mathbf{p}} = \text{avg}(h\sqrt{u^2+v^2})$ (m^{2}s^{-1})
$n = 0.01$	$4.812 \times 10^{-3} + 3.540 \times 10^{-1}$	7.315×10^{-7}	9.575×10^{-3}
$n = 0.045$	$4.610 \times 10^{-3} + 3.474 \times 10^{-1}$	7.177×10^{-7}	9.371×10^{-3}
$n = 0.1$	$3.828 \times 10^{-3} + 3.230 \times 10^{-1}$	6.663×10^{-7}	8.536×10^{-3}

Table 5.3: Energies and velocity calculated by *Riemann2D* for different values of Manning's n at time $t = 3$ hours. The first column shows added calculated energy per unit area, made of kinetic energy + potential energy. The second shows energy added as a proportion of original energy. The third column shows the average fluxes.

5.1.2 Effect of varying bed friction

Three tests were done to determine the effect of Manning's n, or bed friction. The finest mesh was used (23192 elements) with a wind speed of $2\,\text{ms}^{-1}$ imposed in a westerly direction, minDepth $= 1 \times 10^{-6}$ m, a fixed timestep dt $= 1$ s, and the values of Manning's n used were 0.1, 0.045 and $0.01\,\text{m}^{-1/3}\text{s}$. According to Chow (1959), $n = 0.01$ corresponds to brass or smooth metal, $n = 0.045$ to weeds and rocks, and $n = 0.1$ to dense brush, so these values are all in a physically realistic range. Coriolis was zero so as not to affect the flow behaviour. The time taken to run the simulation was around 65 minutes. The energy added per unit area, the rate of added energy, and average velocity are shown in Table 5.3.

A higher value of Manning's n leads to more energy being dissipated via a rougher bed and flows are slower. As shown in Chapter 3, even at its maximum realistic value ($n^2 = \mathcal{O}(\varepsilon^3)$ [1]), the presence of Manning's n in the equation leads to effects that are visible only at $\mathcal{O}(\varepsilon^4)$ or smaller. Table 5.3 shows a variation in the fourth decimal place for the value of n^2 changing from $\mathcal{O}(\varepsilon^4)$ to $\mathcal{O}(\varepsilon^3)$, agreeing with what was shown in Chapter 3.

Because such a simple domain is used here, it allows these tests to show very clearly how varying parameters such as wind speed, mesh density and bed friction, leads to realistic changes in the solution, and the changes are almost exactly whole orders of magnitude.

1. Although we considered $n^2 = \mathcal{O}(\varepsilon^2)$, this would never occur in the circumstances we are studying here *i.e.* slow, everyday flows. Such a high value would only be seen in a flood situation when water flows over land populated with trees. The value of $n^2 = \mathcal{O}(\varepsilon^2)$ is nevertheless useful for theoretical or demonstrative purposes.

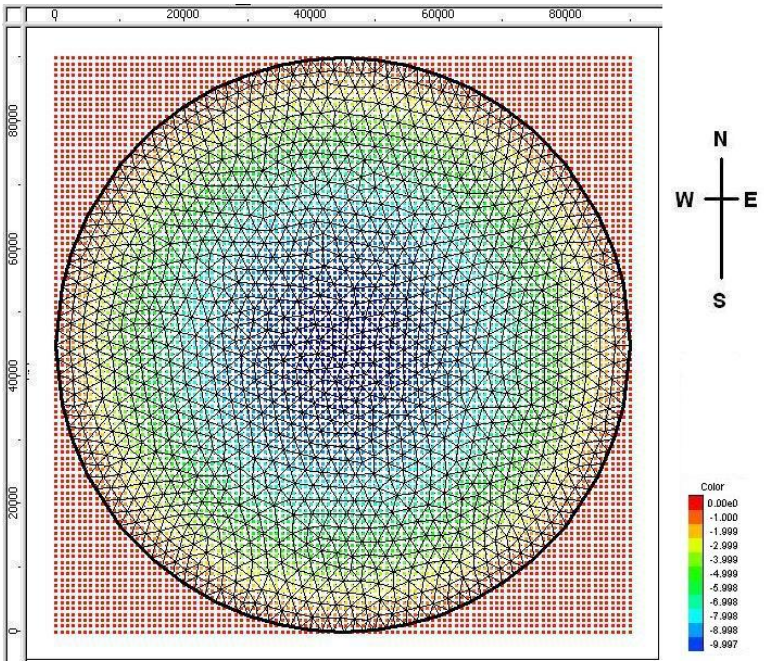


Figure 5.2: Mesh of 2484 elements of a circular domain of 45 km radius and maximum depth of 10 m.

5.2 Parameter variations in a circular basin

The tests that were carried out for the square were repeated for the circular basin; in other words, variation of parameter values of the wind speed and Manning's n. The circular basin-like domain had a bottom topography described by

$$b = b_0 \left(1 - \frac{x^2 + y^2}{r_0^2}\right),$$

where b_0 is the maximum bed elevation, and r_0 is the radius. The domain is partly inspired from Port Phillip bay, Melbourne, as it is roughly circular, so we take for the dimensions of the circular basin-like domain that the radius of the bay is $r_0 = 4.5 \times 10^4$ m, with a maximum depth $b_0 = -10$ m (negative in the z-direction).

A mesh of 21996 elements was used: a mesh with one tenth that number (2484 elements) is shown in Figure 5.2 as the denser mesh would not allow the triangular elements to be seen easily.

Because of the larger scope for instability due to the vanishing depth of the bathymetry, the value of minDepth was set to be 1×10^{-2} m. This larger value ensures no numerical instabilities develop in the flow due to the vanishingly small water depth.

Due to varying bathymetry, the numerical results that follow are quite different from the simple square domain that has flat bathymetry. When comparing tests for the two domains, notice that the flow velocities or the energy do not vary by complete orders of magnitude for each corresponding variation in wind speed in the case of the basin, as does happen for the square.

For all tests, the code was run for a simulation time of three hours, which took around 58 minutes, and the Courant number was around 0.0195.

5.2.1 Effect of different wind speeds

The effect of different wind speeds was tested by simulating wind blowing across the surface of the bay in a westerly direction for three hours, once at a wind speed $12\,\mathrm{ms}^{-1}$, once for $1.2\,\mathrm{ms}^{-1}$ and once for $0.12\,\mathrm{ms}^{-1}$. These lower wind speeds are chosen to prevent numerical instabilities from forming in areas where a stronger wind leaves a very shallow depth of water (it was also necessary to set the value of minDepth $= 0.2$ m for the case of $w = 12\,\mathrm{ms}^{-1}$). The timestep was fixed at dt $= 1$ s, the wind friction coefficient was 1×10^{-3}, and Manning's n and Coriolis were zero.

The total initial (potential) energy was calculated to be 1.0394×10^{15} J, or $E_0 = 1.643 \times 10^5$ J per unit area, which is almost exactly what *Riemann2D* found (1.0382×10^{15} J), the differences being due to the way the boundary of the circle is approximated by straight sections.

Table 5.4 shows that average flux varies by between one and three orders for a variation in wind speed of one order. From the analysis of Chapter 3, we would expect it to vary by two orders, so the shape of this domain, which is what distinguishes this test from the equivalent test on the square domain, is responsible for this deviation.

From the information in Table 5.4 we see that average velocity is slightly higher than for the square. The energy gained by the circle is greater than the energy gained for the square, even with the lower wind speed. This is mostly due to kinetic energy, which here contributes as much as potential energy, because of the areas of shallower water that are more subject to

wind speed	$\Delta E = \Delta KE + \Delta PE$ (J = kg m^{2}s^{-2})	$\Delta E/E_0$	$\tilde{\mathbf{p}} = \text{avg}(h\sqrt{u^2+v^2})$ (m^{2}s^{-1})
$w = 12$	$0.937 \times 10^3 + 1.904 \times 10^3$	1.729×10^{-2}	1.879×10^1
$w = 1.2$	$2.209 \times 10^{-1} + 1.676 \times 10^{-1}$	2.365×10^{-6}	2.048×10^{-2}
$w = 0.12$	$2.235 \times 10^{-5} + 5.425 \times 10^{-6}$	1.691×10^{-10}	2.050×10^{-4}

Table 5.4: Energies and flow velocities calculated by *Riemann2D* for different values of w at time $t = 3$ hours. The first column shows total calculated energy per unit area made up of added energy + added potential energy, the second column shows energy added as a proportion of original energy, and the third shows the average fluxes.

friction. The rate of energy added, $\Delta E/E_0$, is only a little lower than the square, despite the wind speed being nearly half. This shows the varying bathymetry is much more susceptible to energy effects. We also see that $\Delta E \ll E_0$, which agrees with what was found in Chapter 3.

5.2.2 Effect of varying bed friction

A test was done to investigate the effect of varying bed friction. The value of Manning's n was varied in tests for $n = 0.1$, $n = 0.045$ and $n = 0.01\,\text{m}^{-1/3}\text{s}$ for a simulation time of three hours. The wind speed was $1.2\,\text{ms}^{-1}$, the fixed timestep was dt $= 1\,$s and the wind friction coefficient 1×10^{-3}.

Table 5.5 shows that the average flux for the circular basin is almost three orders higher in the circle than in the square basin, despite wind speed being lower. This shows that inaccuracies in bathymetric data may lead to spurious velocities when making simulations of a 'real' bay of arbitrary bathymetry, which contains areas of very shallow water, since flow velocities are sensitive to variations in bathymetry, .

Table 5.5 also shows that by altering n^2 from $\mathcal{O}(\varepsilon^4)$ to $\mathcal{O}(\varepsilon^3)$ results in a small correction in the fourth decimal place to the velocity. This agrees with what was shown in the analysis of Chapter 3, that even at its maximum value, Manning's n affects the flow only at low orders.

Manning's n	$\Delta E = \Delta KE + \Delta PE$ (J = kg m^{2}s^{-2})	$\Delta E/E_0$	$\tilde{\mathbf{p}} = \text{avg}(h\sqrt{u^2+v^2})$ (m^{2}s^{-1})
$n = 0.01$	$8.083 \times 10^{-2} + 1.885 \times 10^{-1}$	1.639×10^{-6}	1.816×10^{-2}
$n = 0.045$	$2.624 \times 10^{-2} + 1.907 \times 10^{-1}$	1.320×10^{-6}	1.334×10^{-2}
$n = 0.1$	$1.065 \times 10^{-2} + 1.732 \times 10^{-1}$	1.121×10^{-6}	0.919×10^{-2}

Table 5.5: Energies and velocity calculated by *Riemann2D* for different values of Manning's n at time $t = 3$ hours. The first column shows added calculated energy per unit area, made of kinetic energy + potential energy. The second shows energy added as a proportion of original energy. The third column shows the average fluxes.

5.3 Axisymmetric, steady-state solutions

Two important analytical results are available for flow in a circular basin-like domain, and we study both in this section. The analytical results are then compared to simulations in *Riemann2D* as part of the validation process that provides the confidence in the numerical results.

5.3.1 Shallow water equations in a circular basin

One of the few analytical results available from the shallow water equations for slow flows is that of rotating steady-state flow in a circular basin. As such it can be used to give comparisons with numerical results, which is useful for validation of *Riemann2D* .

Consider a circular, basin-like domain where the bathymetry is described in cylindrical polar coordinates by

$$b = b_0 \left(1 - \frac{r^2}{r_0^2}\right), \tag{5.2}$$

where b_0 is the maximum depth, r is the radial coordinate, r_0 is the radius of the domain, the level of still water is $z = 0$ and in the region $z > 0$ the sides of the domain are vertical, as shown in Figure 5.3.

We take $r_0 = 4.5 \times 10^4$ m, and $b_0 = -10$ m, as before. We denote by h the total depth of the water so $h + b$ is the perturbation from still water, the radial velocity is $\dot{r} = u$ and the axial

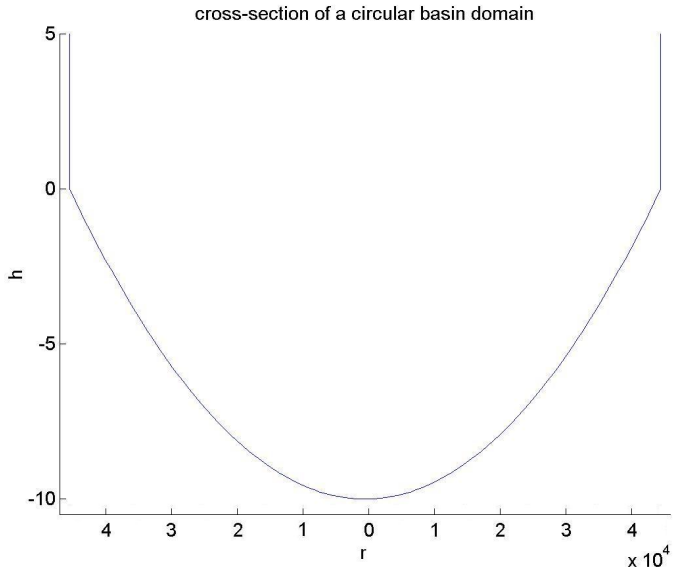


Figure 5.3: Visualisation of domain considered in this problem – the cross-section of a circular basin of radius $r_0 = 4.5 \times 10^4$ m, and maximum depth $b_0 = -10$ m.

is $r\dot{\theta} = v$. The gravitational constant is g, as usual.

The full (r, θ)-version of shallow water equations in cylindrical polars can be derived from the depth-averaged Navier-Stokes equations (see Whitham (1974), Acheson (1990), Hafez and Dimanlig (1996)). They are

$$\frac{\partial h}{\partial t} + \frac{1}{r}\frac{\partial (rhu)}{\partial r} + \frac{1}{r}\frac{\partial (hv)}{\partial \theta} = 0 \tag{5.3a}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial r} + \frac{v}{r}\frac{\partial u}{\partial \theta} - \frac{v^2}{r} + g\frac{\partial h}{\partial r} = -g\frac{\partial b}{\partial r} \tag{5.3b}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial r} + \frac{v}{r}\frac{\partial u}{\partial \theta} + \frac{uv}{r} + \frac{g}{r}\frac{\partial h}{\partial \theta} = -\frac{g}{r}\frac{\partial b}{\partial \theta}, \tag{5.3c}$$

where we neglect bed friction, wind stress and Coriolis. Here, $h = h(r, \theta)$ is the depth of water, $u = u(r, \theta)$ is the radial velocity, $v = v(r, \theta)$ is the axial velocity. The right hand side represents bedslope, since $-b$ (given by (5.2)) is the depth of the domain from a fixed reference level.

We assume the water to be spinning in the domain in a steady way, with no outward movement, *i.e.* $u = 0$, and a constant angular velocity, *i.e.* $\dot{\theta} = \kappa$ so that $v = \kappa r$ for some constant κ. The solution h is therefore independent of θ. Under these assumptions the system

(5.3) reduces to

$$-\kappa^2 r + g\frac{\partial h}{\partial r} = -g\frac{\partial b}{\partial r},$$

the first and third equations being satisfied trivially. Integrating, we obtain the equation for the water profile as

$$h = -b + \int \frac{\kappa^2 r}{g}\,\mathrm{d}r + B, \tag{5.4}$$

for some constant B that can be taken to be zero without loss of generality. In the case where κ is a constant, the velocity profile and the streamlines coincide. This also implies that vorticity ($\omega = \nabla \wedge \dot{\mathbf{r}} = 2\dot{\theta}$) is constant.

We require that conservation of mass is satisfied: the volume of fluid must be the same when the fluid is spinning as when it is still.

The volume of water under the curve described by equation (5.4) is equal to the volume under the curve describing still water, $h = -b$. Thus conservation of mass means

$$\int_0^{r_0} -b + \frac{\kappa^2 r^2}{2g} + A\,\mathrm{d}r = \int_0^{r_0} -b\,\mathrm{d}r,$$

which determines the constant A as

$$A = -\frac{\kappa^2 r_0^2}{6g}.$$

Hence the equation of the free surface in radial coordinates is the parabola

$$h = -b + \frac{\kappa^2}{2g}\left(r^2 - \frac{r_0^2}{3}\right), \tag{5.5}$$

and in Cartesian coordinates this is

$$h = -b_0\left(1 - \frac{x^2+y^2}{r_0^2}\right) + \frac{\kappa^2}{2g}\left(x^2 + y^2 - \frac{r_0^2}{2}\right).$$

Since physically, we must have $h > 0$, this puts a constraint on the rate at which the water is spinning: the level of water is at its lowest at $r = 0$, so (5.5) implies $\kappa < \sqrt{-6b_0 g/r_0^2}$ (recall that $b_0 < 0$). Thus the slower the water is spinning round, the larger the radius that admits valid solutions.

Based on the dimensions being used, we require that the velocity $v < 5.39r \times 10^{-4}\,\mathrm{ms}^{-1}$ for the validity of the analytic solution to hold. The surface elevation for the cases with $\kappa = 1.5 \times 10^{-4}\,\mathrm{s}^{-1}$ and $\kappa = 3 \times 10^{-4}\,\mathrm{s}^{-1}$ are shown in Figures 5.7 and 5.8, respectively.

These analytical results are compared with the numerical results later to test how well *Riemann2D* deals with solid body rotation.

5.3.2 Shallow water in an elliptical basin

In the second part to this problem we consider that the boundary of the domain is perturbed to be an ellipse. Instead of solving the shallow water equations as in the previous section, we use Bernoulli's principle for an ideal fluid and conservation of mass, and make use of the fact that, for a steady state problem, the streamlines and the velocity profile coincide. The advantage with this approach is that we use Cartesian coordinates for the majority of the calculation.

The equation for an ellipse is

$$\left(\frac{x}{\alpha}\right)^2 + \left(\frac{y}{\beta}\right)^2 = 1,$$

where we choose $\alpha = r_0(1+\delta)$, $\beta = r_0(1-\delta)$, r_0 is the radius, and $\delta > 0$. Note that when $\alpha = \beta$, this equation reduces to that of a circle.

We denote by h the total depth of water, b the bathymetry of the domain from a fixed reference point (thus $h + b$ is the perturbation from still water), $u = \dot{x}$ the velocity in the x-direction, $v = \dot{y}$ the velocity in the y-direction. The equation for the bathymetry of the elliptical basin is

$$b = -b_0\left(1 - \left(\frac{x}{\alpha}\right)^2 + \left(\frac{y}{\beta}\right)^2\right)$$

in the region $z < 0$, and the sides of the domain are vertical in the region $z > 0$.

Streamlines follow the shape of the boundary *i.e.* they are all concentric ellipses. Therefore the streamlines ψ are described by

$$\psi = \frac{Q(r_0)}{r_0}\left[\left(\frac{x}{\alpha}\right)^2 + \left(\frac{y}{\beta}\right)^2\right],$$

for some discharge function $Q(r_0)$ to be determined. We also have that, in an ideal fluid,

$$u = \frac{\partial\psi}{\partial y}, \qquad v = -\frac{\partial\psi}{\partial x}$$

thus

$$u = \frac{2y}{r_0\beta^2}Q(r_0), \qquad v = -\frac{2x}{r_0\alpha^2}Q(r_0). \tag{5.6}$$

To determine Q we must take a brief detour into polar coordinates. In elliptic polars, $x = \alpha r\cos\theta$, $y = \beta r\sin\theta$. We assume no outward velocity, so $\dot{r} = 0$, and constant angular velocity, so $\dot{\theta} = \kappa$ for some constant κ. Therefore

$$(u,\, v) = (-\alpha r\kappa\sin\theta,\, \beta r\kappa\cos\theta)\,. \tag{5.7}$$

Equations (5.6) and (5.7) together imply $Q(r_0) = \frac{1}{2}\alpha\beta r_0\kappa$. Now

$$\mathbf{u}^2 = u^2 + v^2 = \alpha^2\beta^2\kappa^2\left(\frac{x^2}{\alpha^4} + \frac{y^2}{\beta^4}\right).$$

We wish to determine h, so we make use of Bernoulli's principle, which says that for an ideal fluid

$$h + b - \frac{\mathbf{u}^2}{2g} = A$$

along streamlines, for some constant A. The principle of conservation of mass determines the unknown A. We require

$$\int \frac{\mathbf{u}^2}{2g} + A\,\mathrm{d}x\,\mathrm{d}y = 0,$$

which means

$$\Rightarrow \quad A = -\frac{\kappa^2}{8g}(\alpha^2 + \beta^2).$$

Thus the free surface is given by

$$h = -b + \frac{\kappa^2}{2g}\left(\frac{\beta^2}{\alpha^2}x^2 + \frac{\alpha^2}{\beta^2}y^2 - \left(\frac{\alpha^2+\beta^2}{4}\right)\right). \tag{5.8}$$

Note that when $\alpha = \beta = r_0$, (5.8) becomes the same solution as for a circular domain, (5.5).

Plots in Figures 5.9-5.10 show the surface elevation given by (5.8) in the x- (top) and y- (bottom) planes. The radius shown is $r_0 = 4.5\times10^4\,\mathrm{m}$, $\delta = 0.1$, and the angular velocities are with $\kappa = 1.5\times10^{-4}$ and $\kappa = 3\times10^{-4}$. These analytical results are compared with the numerical solution from *Riemann2D* in the next section.

5.3.3 Numerical results

A test for solid body rotation was carried out to compare with the analytical results (5.5) and (5.8) given above. The domains used were a circular basin of 610 elements (bathymetry given

by (5.9)) and an elliptical basin of 606 elements (bathymetry given by (5.10)). The meshes used are shown in Figure 5.4.

The bathymetry of the circular basin was b, given by

$$b = b_0 \left(1 - \frac{r^2}{r_0^2}\right), \tag{5.9}$$

with $b_0 = -10\,\mathrm{m}$, $r_0 = 45\,\mathrm{km}$ and the surface profile was set as

$$\eta = \frac{\kappa^2}{2g}\left(x^2 + y^2 - \frac{r_0^2}{2}\right), \quad \text{with} \quad u = \kappa y, \quad v = -\kappa x,$$

for $\kappa = 1.5 \times 10^{-4}$, $\kappa = 3 \times 10^{-4}$. For the elliptical basin, the bathymetry was

$$b = -b_0 \left(1 - \left(\frac{x}{\alpha}\right)^2 + \left(\frac{y}{\beta}\right)^2\right), \tag{5.10}$$

where $\alpha = r_0(1+\delta)$, $\beta = r_0(1-\delta)$, r_0 is the radius, and $\delta = 0.1$ and the surface profile was fixed as

$$\eta = \frac{\kappa^2}{2g}\left(\frac{\beta^2}{\alpha^2}x^2 + \frac{\alpha^2}{\beta^2}y^2 - \left(\frac{\alpha^2+\beta^2}{4}\right)\right), \quad \text{with} \quad u = \frac{\alpha}{\beta}\kappa y, \quad v = -\frac{\beta}{\alpha}\kappa x,$$

with $\kappa = 1.5 \times 10^{-4}$, $\kappa = 3 \times 10^{-4}$. The surface profiles were set initially in the meshfile, then *Riemann2D* was run for a simulation time of one hour, updating only the surface elevation at each timestep (*i.e.* not the velocities) in order to test how well the surface is preserved under rotation at a fixed speed. The boundary condition type was reflective. Two values of angular velocity were tested with κ being $\kappa = 1.5 \times 10^{-4}$ and $\kappa = 3 \times 10^{-4}$. The timestep used was fixed with dt $= 10\,\mathrm{s}$, the roughness coefficient used was Manning's $n = 0.03\,\mathrm{m}^{-1/3}\mathrm{s}$, the value of minDepth was $1 \times 10^{-6}\,\mathrm{m}$ and the Courant number was around 0.0028. The tests were performed once with no limiter and once again with the MLG limiter. Tests with no limiter took around 15 seconds to run, those with the MLG limiter took around 28 seconds. Results for the circle are shown in Figures 5.7 and 5.8. Results for the ellipse are shown in Figures 5.9 and 5.10.

It was shown by Jha (2006) that tests carried out with the MLG limiter suffered much less numerical diffusion than no limiter or even the limiters such as minmod, superbee, LCD or the extended Van Leer (see Figure 5.6, which although gives results from a different type of test, shows clearly how effective the MLG limiter is at preventing numerical diffusion). Numerical

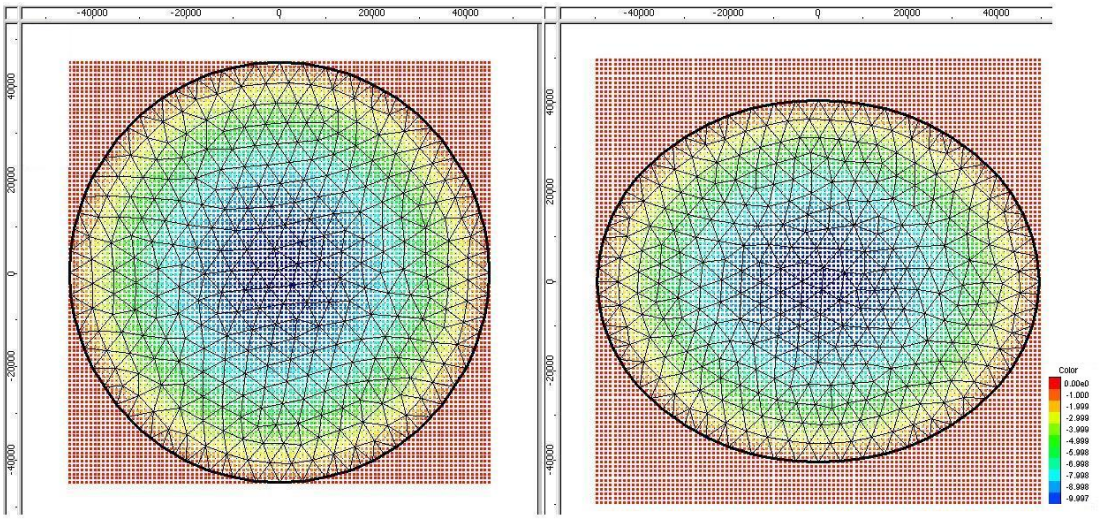


Figure 5.4: Meshes used for solid-body rotation tests. Left shows the circular domain of 610 elements, right shows ellipse of 606 elements.

results where no limiter was used (Figures 5.7, top and 5.8 top, Figures 5.9 first and third and 5.10 first and third) show poor agreement with analytical solutions, and are badly smeared. The same tests using the MLG limiter clearly have much better agreement with the analytical results.

Figure 5.6a) suggests that at least 2000 elements should ideally be used to fully capture the effect of rotation, supporting the knowledge that a finer mesh displays less numerical diffusion than a coarser one. Nevertheless, our meshes of approximately 600 elements show very clearly the effect of a good limiter, and this is because these tests are considerably less demanding than the tests done by Jha (2006).

This is a very valuable test that demonstrates how well *Riemann2D* reproduces an analytical solution. The very close results provide confidence in *Riemann2D* and the improvement in the results by using a limiter means that in the simulations in the rest of this thesis will always use the MLG limiter.

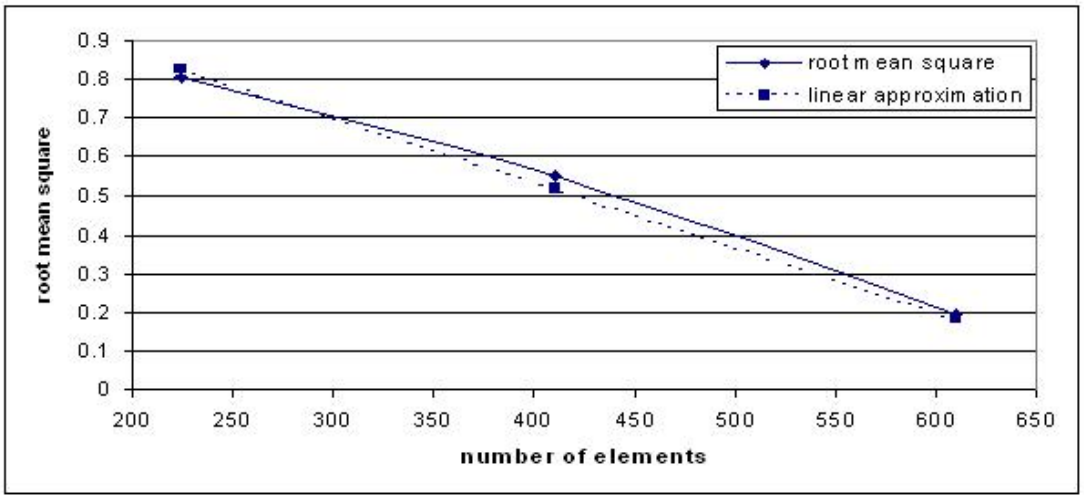


Figure 5.5: Plot of root mean square to number of elements. As the number of elements is increased, the error reduces. The linear equation that approximates the results is also shown (dotted line).

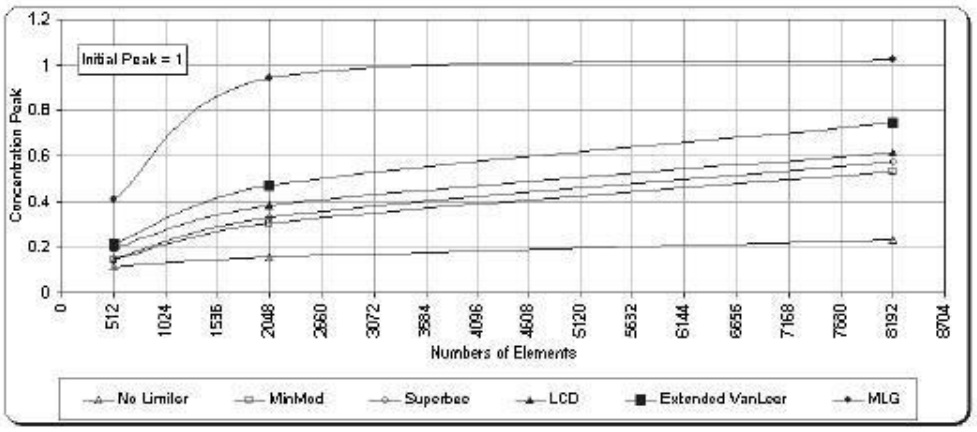
5.3.4 Convergence of *Riemann2D*

This analytical solution allows us to test the convergence of *Riemann2D* . As the mesh of calculation is refined, we expect the numerical solution to approach the analytical. To this end, the test using the circular domain, with the MLG limiter and $\kappa = 3 \times 10^{-4}$ was repeated on meshes of 410 elements and 224 elements. The average differences between the simulated and the actual solution are found by taking the root mean square.

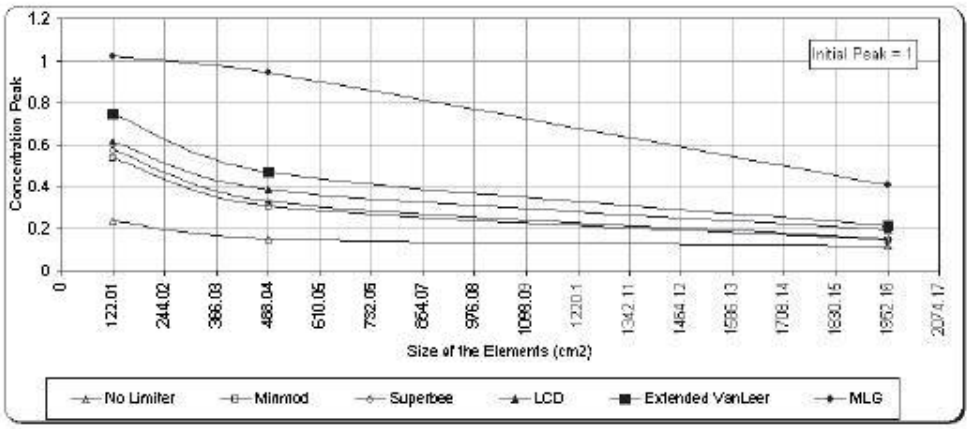
This error can be approximated by the linear equation

$$\text{RMS} = \frac{\text{no. elements}}{-600} + 1.2$$

(see Figure 5.5). So as the number of elements in the computational mesh increases, the value of the error decreases. We have satisfied the convergence, the $\mathcal{C}$ property, and since the CFL condition is also always satisfied, we can use *Riemann2D* with confidence.



(a)



(b)

Figure 5.6: Graph to show effectiveness of different limiters. Reproduced from Jha (2006). The test carried out was a rotation of a concentration of initial value 1. After one rotation, the concentration is shown here (Figure (a)) for no limiter and the five limiters minmod, superbee, LCD, Extended Van Leer and MLG. Ideally, the concentration has value one after one rotation. It can be seen that the MLG limiter performs best. Figure (b) shows that as the area of the elements in the mesh increases, results are smeared even with a good limiter. This is related to the CFL condition for convergence.

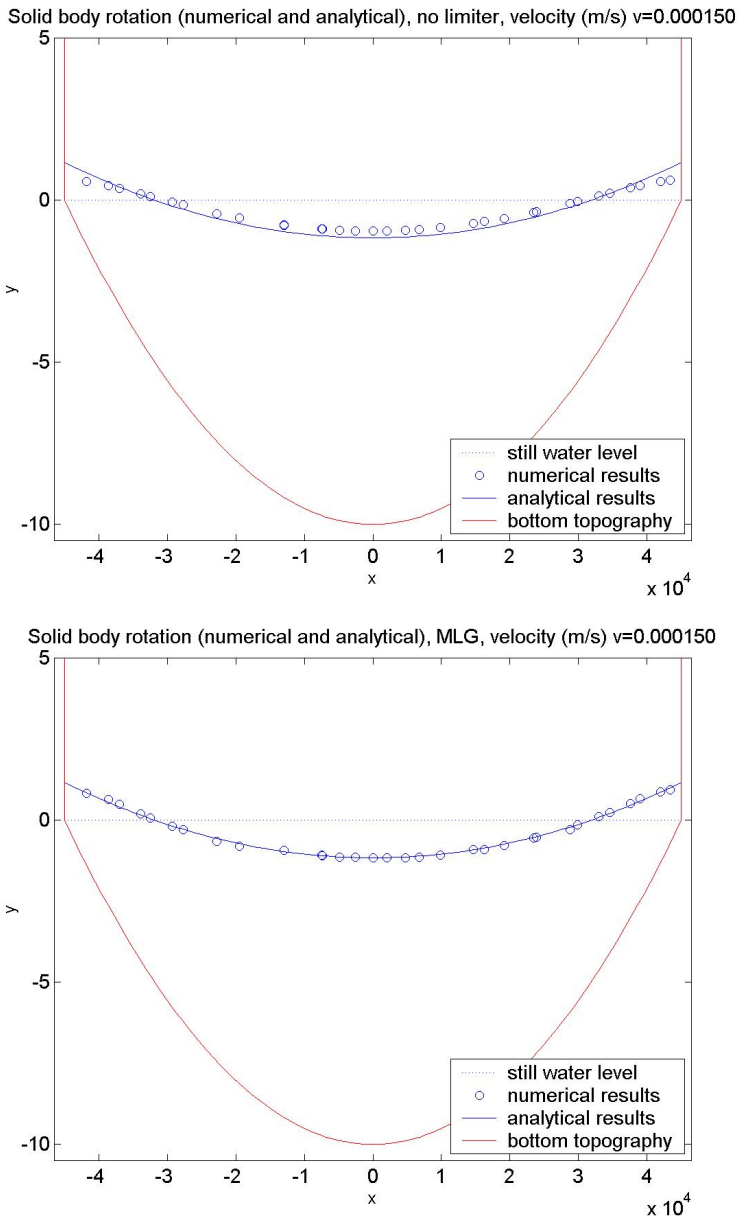


Figure 5.7: Effect of solid body rotation on a spherical dish. Results were obtained from running *Riemann2D* for a time of one hour on a mesh of 610 elements. Analytical results are shown (solid line), numerical results are shown as circles. Top figure shows results with no limiter, and the numerical diffusion is clearly visible. Bottom figure shows results with the MLG limiter, showing much closer agreement.

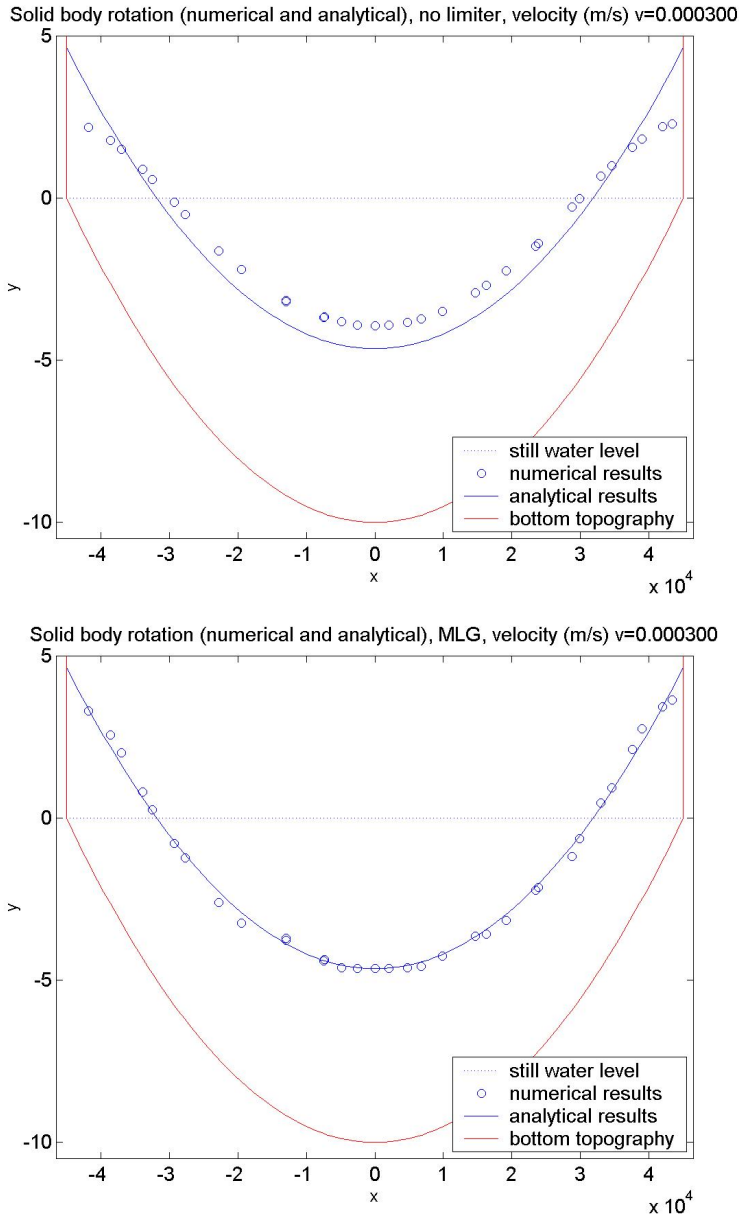


Figure 5.8: Effect of solid body rotation on a spherical dish. Results were obtained from running *Riemann2D* for a time of one hour on a mesh of 610 elements. Analytical results are shown (solid line), numerical results are shown as circles. Top figure shows results with no limiter, and the numerical diffusion is clearly visible. Bottom figure shows results with the MLG limiter, showing much closer agreement.

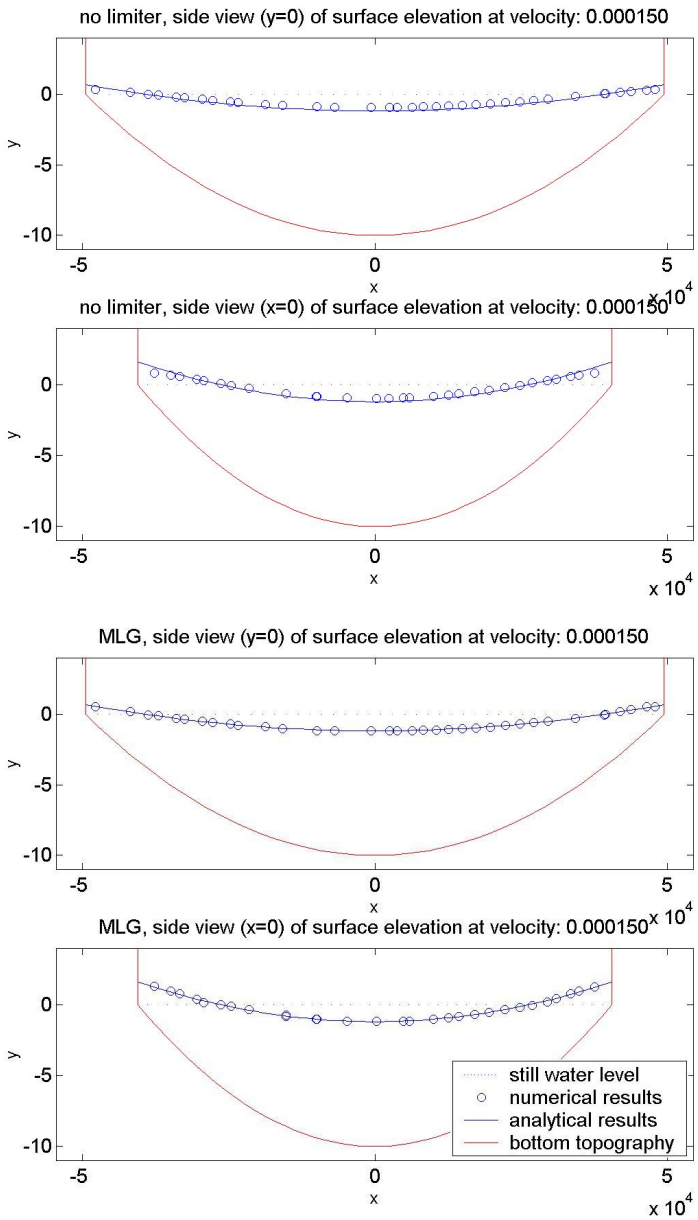


Figure 5.9: Effect of solid body rotation in an elliptical dish. Angular velocity was with $\kappa = 1.5 \times 10^{-4}$. Solid line shows analytical solution, dots show numerical solution. The top two figures (x-plane, y-plane) show results with no limiter, where smearing of the results is visible. The bottom two figures (x-plane, y-plane) show results with the MLG limiter, and results are much closer to the analytical.

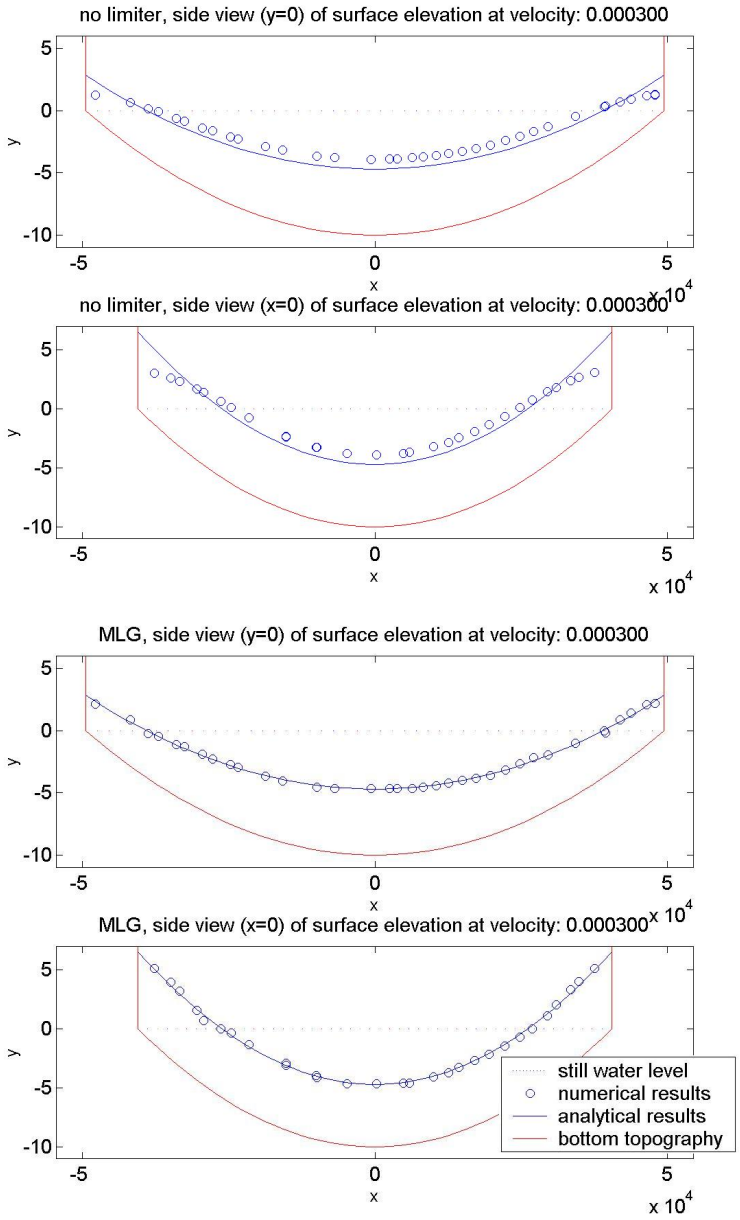


Figure 5.10: Effect of solid body rotation in an elliptical dish. Angular velocity was with $\kappa = 3 \times 10^{-4}$. Solid line shows analytical solution, dots show numerical solution. The top two figures (x-plane, y-plane) show results with no limiter, where smearing of the results is visible. The bottom two figures (x-plane, y-plane) show results with the MLG limiter, and results are much closer to the analytical.

5.4 Steady-state wind in a circular basin

A test was carried out to attempt to reproduce the numerical results of Borthwick *et al.* (2001a) and Borthwick *et al.* (2001b) and the analytical results of Kranenburg (1992). In this test, a circular basin defined by (5.2), but this time with radius $r_0 = 192$ metres and a maximum depth at the centre of $b_0 = -1$ metre. Wind of speed $10\,\mathrm{ms}^{-1}$ with a wind friction coefficient of 2×10^{-3} was blown over the initially flat surface in a westerly direction until a steady state was achieved (one hour). The Courant number was around 0.27 and the timestep was fixed at dt $= 1$ s. Figure 5.11 (top) shows the mesh used with 623 elements and Figure 5.11 (bottom) shows the steady-state velocity field. The results are, visually, virtually identical to those published by Borthwick *et al.* (2001a) and Borthwick *et al.* (2001b), showing two areas of recirculation, with velocity vectors pointing in the direction of the wind at the edge, and in the opposite direction in the centre. This velocity profile occurs because shallower water is more subject to the shear effect from the wind, and hence flows in the same direction. Since mass is conserved in the domain, there must be some recirculation, and this happens in the region where the wind's shearing effect is lowest, at the centre. The 'clustering' of arrows in the numerical results is due to the smaller elements that Argus generated at the top (north) that can be seen in the top figure.

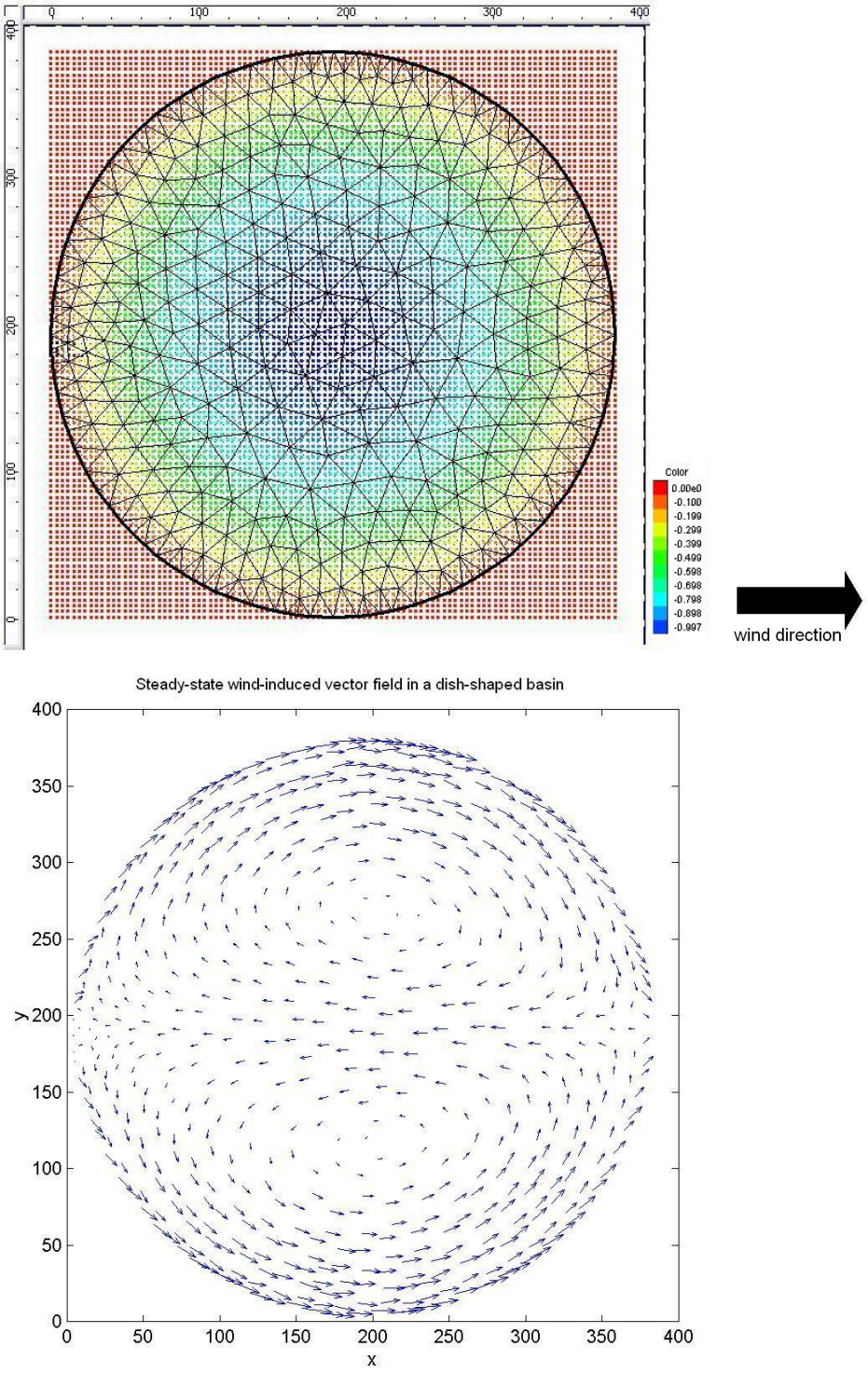


Figure 5.11: Steady-state problem of wind of $10\,\mathrm{ms}^{-1}$ on circle of 192 m radius, maximum depth 1 m and 623 elements. The results show two areas of recirculation, which is in good qualitative agreement with results of Borthwick *et al.* (2001a) and Borthwick *et al.* (2001b)

5.5 Summary

In this chapter we have

- *Carried out tests on square and circular basin domains in which parameter values for source terms were varied. The different sizes of source terms were seen to affect the solution in the expected way, based on the work from previous chapters.*
- *Demonstrated that both increasing wind speed and decreasing bed friction have more effect in adding energy to the circular basin than to the square. This is due to the shallower areas in the circular basin that are more subject to friction and wind shear, generating more kinetic energy.*
- *Demonstrated convergence of the scheme: that the simulated result approaches the analytical as the mesh is refined.*
- *Tested solid body rotation in shallow circular and elliptical basins and shown that results match well with the analytical results. This test also demonstrated the importance of limiters for such flows.*
- *Simulated wind blowing over a shallow circular basin, and used a published numerical result (Borthwick et al. (2001a) and Borthwick et al. (2001b)) and a well-known analytical result (Kranenburg (1992)) against the Riemann2D code, finding excellent agreement.*

This chapter has shown that the model we are using produces the correct physical behaviour, and that *Riemann2D* is accurate.

In the next chapter the same domains are used to validate the time-dependent (periodic) boundary condition that is intended to replicate the tide. The combined effects of wind and tides on the flow pattern and on the path of a contaminant that is released into the bay will be studied.

CHAPTER 6

Tidal simulation and pollutant transport

In this chapter we use the square and the circular basin domains of the previous chapter to test the time-dependent boundary condition described in Chapter 4 and the effect it has on a passive pollutant.

The periodic boundary condition is intended to simulate tidal conditions. It has parameters (amplitude, mean, frequency) that can be chosen to most closely resemble the tide in question. Since we are considering generic models in this chapter, we choose these parameters to be based on average typical values. So for periodic boundaries, the state value η in the ghost cell was set as

$$\eta_{ghost} = m + a\sin(ft + \varphi). \tag{6.1}$$

The mean, m and the phase, φ can be taken as zero without loss of generality; for the amplitude, a, we choose 1.25 m, as the tidal range for Moreton Bay is 0.9 m to 1.99 m at spring and neap tides respectively (see Chapter 7, Figure 7.3); and the frequency, $f = 2\pi/T$ where $T = 24$ hours as this corresponds to a semi-diurnal tide, not only more common in the world, but the only type of tide seen on the eastern Australian coast. The boundary conditions for hu, hv were reflective (fluxes are zero on the boundary), meaning no Riemann problem is solved for these variables at the boundary.

6.1 Time-dependent boundary conditions (square)

The square domain of side 100 km and constant depth 10 m was used to test the tide-like time-dependent boundary condition with medium-density mesh of 2326 elements, periodic-type boundary conditions on the south side, and reflective-type boundary conditions on the other three sides.

The test was run for a simulation time of 10 days before recording any data to allow phenomena related to the initial steady state to be completely dissipated. The fixed timestep was of size dt $= 1\,\mathrm{s}$, Manning's $n = 0.03\,\mathrm{m}^{-1/3}\mathrm{s}$, and minDepth was $1 \times 10^{-6}\,\mathrm{m}$. In all tests for the square, the Courant number was around 0.005, and the simulations took around 9 hours 30 minutes to run.

6.1.1 Comparing Coriolis force

To demonstrate the effect of Coriolis force, the test was run once with the Coriolis parameter taking the value zero (Figure 6.2), and once with Coriolis parameter being $-6.6 \times 10^{-5}\,\mathrm{s}^{-1}$ (Figure 6.3a)-c)), corresponding to a latitude of 27°S, the latitude of Moreton Bay. The arrows in the plots are scaled by a factor of 3, with the average u and v velocities given in Tables 6.1 and 6.2 for the cases without and with Coriolis, respectively. From these tables, we see that the velocity in the y-direction ($\tilde{v}$ values) are largely similar in both cases. The $\tilde{u}$ values, however, are at least an order of magnitude smaller in the case without Coriolis. This agrees with the quasi-1D flow profile in Figure 6.2 and the clockwise flow seen in Figures 6.3a)-c).

Without Coriolis force, the flow is quasi-1D (Figure 6.2). In the southern hemisphere, (see Figures 6.3a)-c)), flow is pulled faster on the west so that water flowing into the bay (at 3, 15 hours and at 12, 24 hours) flows clockwise and water flowing out (at 6, 18 hours) flows anti-clockwise. At mid ebb (6, 18 hours) and mid flood tides (12, 24 hours), there is one area of clockwise flow and one of anti-clockwise because of the time it takes for the tide to influence the water in the bay.

The periodic nature of the boundary condition can be verified by noticing the similarity of average velocities in Table 6.2 at pairs 3 and 15 hours (high tide), 6 and 18 hours (mid ebb), and 12 and 24 hours (mid flood).

time (hours) + 10 days	$(\bar{u}, \bar{v})$ ms^{-1}	time (hours) + 10 days	$(\bar{u}, \bar{v})$ ms^{-1}
3	-2.71×10^{-5}, 0.5537	15	-1.07×10^{-5}, 0.5538
6	-0.0003, 0.0073	18	-0.0002, 0.0071
12	0.0003, -0.1057	24	0.0003, -0.1057

Table 6.1: Average (mean) velocities in the u and v directions relating to Figure 6.2 (no Coriolis). Note the similarity of plots and average velocities at pairs 3 and 15 hours (high tide), 6 and 18 hours (mid ebb), and 12 and 24 hours (mid flood), demonstrating the periodic nature of the boundary condition.

time (hours) + 10 days	$(\bar{u}, \bar{v})$ ms^{-1}	time (hours) + 10 days	$(\bar{u}, \bar{v})$ ms^{-1}
3	-0.0540, 0.5433	15	-0.0539, 0.5433
6	-0.0011, 0.0019	18	-0.0012, 0.0019
12	0.0127, -0.0978	24	0.0128, -0.0977

Table 6.2: Average (mean) velocities in the u and v directions relating to Figures 6.3a)-c) (with Coriolis).

6.1.2 Surface elevation and wind

The same simulation was made twice more, once imposing a westerly wind of $5\,\text{ms}^{-1}$, and once a southerly wind of $5\,\text{ms}^{-1}$. The problem set-up was otherwise unchanged. A wind speed of $5\,\text{ms}^{-1}$ corresponds to a gentle breeze, or 3 on the Beaufort wind scale, which according to the Australian Bureau of Meteorology, is a typical magnitude for Moreton Bay.

The contour plots of the surface elevation and the velocity profiles are given in Figures 6.4a)-c) (westerly wind) and 6.5a)-c) (southerly wind). Compare with plots with no wind in Figures 6.3a)-c). The lack of symmetry in the surface elevation is due to the Coriolis force.

The westerly wind results in a largely unchanged velocity profile. The surface elevation is altered only very little: the water is slightly deeper at mid ebb (6, 18 hours) and slightly shallower at mid flood (12, 24 hours). This is because the wind has a small effect in preventing

the water flowing in or out of the bay.

The southerly wind has a slightly larger effect on the surface elevation contours, pushing the water further into the bay so that the northern part of the bay is slightly deeper than in the case with no wind. In particular, at high tides (3, 15 hours) we see deeper elevation on the west side so that the southerly wind is amplifying the effect of Coriolis. At mid flood tides (12, 24 hours) the elevation is deeper on the east side, as most of the flow is still directed outwards, and hence flowing anti-clockwise so the wind pushes the water to the east side.

These two tests demonstrate quite clearly how wind can affect the surface elevation of the water in this domain, but tidal effects are still dominant and Coriolis remains visible.

6.1.3 Effect of wind direction on pollutants

We consider a pollutant to be a passive tracer that does not affect the flow. As mentioned in Chapter 4, the following equation is solved by *Riemann2D* to model the advection of a pollutant

$$(hc)_t + (huc)_x + (hvc)_y = 0,$$

where c represents the concentration (Periáñez (2004) and Benkhaldoun *et al.* (2007)). The initial concentration in the central four elements on the south side of the bay was set to have a nominal value of $0.3\,\mathrm{kg\ m^{-3}}$, and all other elements had initial value zero. To simulate a fixed concentration, the boundary condition on hc at the tidal boundary was

$$hc_{ghost} = -0.3b,$$

with $b = -10\,\mathrm{m}$, and this was updated at each timestep.

Passive tracer

Scatter plots of the concentration of a passive tracer are shown in Figures 6.6 a)-c). The empty circles indicate elements with a concentration, c, of $0.01 < c < 0.1\,\mathrm{kg\ m^{-3}}$ and the filled circles indicate elements where the concentration is $c > 0.1\,\mathrm{kg\ m^{-3}}$. The contaminant reaches at most 20 km from its origin. At 24 hours it is still slightly affected by the moving tide, suggesting it has not quite reached a steady state. Indeed, Figure 6.1 shows that the concentration in the ten elements nearest the point of contaminant release does not fit a simple sine function, which explains why the contaminant profile at 3 hours is different from the later times.

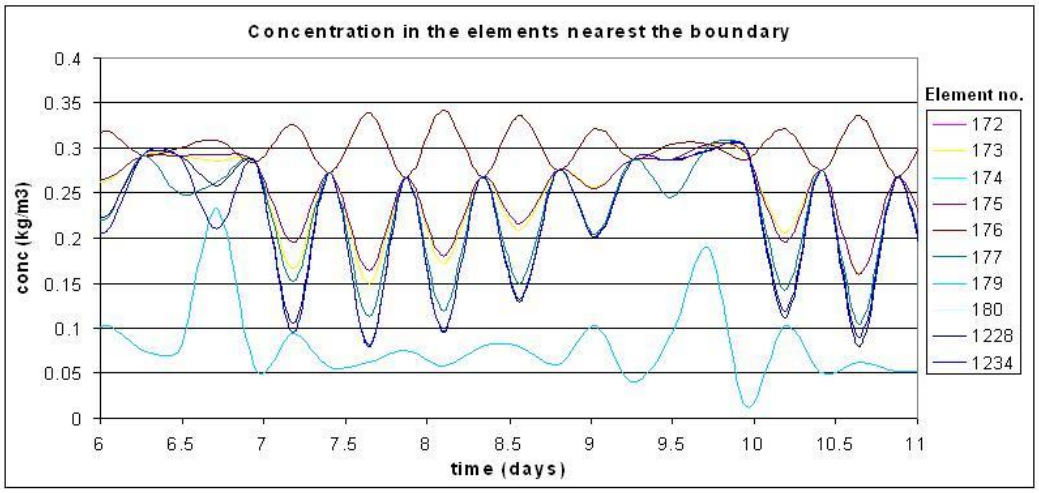


Figure 6.1: Plot of the contaminant in the ten elements nearest to the point of contaminant release. The element numbers correspond to the elements nearest the boundary where pollutant was released. The pattern does not fit a simple sine function, which explains why 3 hours and 15 hours in Figures 6.1 have different profiles.

The effect of Coriolis force is evident from the slight asymmetry in the profile (Figures 6.6a)-c)). Faster flow on the west side means contaminant is pulled in that direction.

Passive tracer with wind

This test was repeated once with a westerly wind of speed $2\,\mathrm{ms}^{-1}$ but the effect on the movement of the contaminant was found to be indistinguishable from the results with no wind.

Scatter plots of the concentration with westerly wind, $5\,\mathrm{ms}^{-1}$, are given in Figures 6.7a)-c). The area of higher concentration does not change significantly but the area of lower concentration is spread out to the east.

Figures 6.8a)-c) show the scatter plots for southerly wind of $5\,\mathrm{ms}^{-1}$. There is almost no difference in the distribution of the pollutant as for the case with no wind. This is because the wind is in the same direction as the tide, so the wind is unable to counteract the tidal effects in the way a westerly wind could.

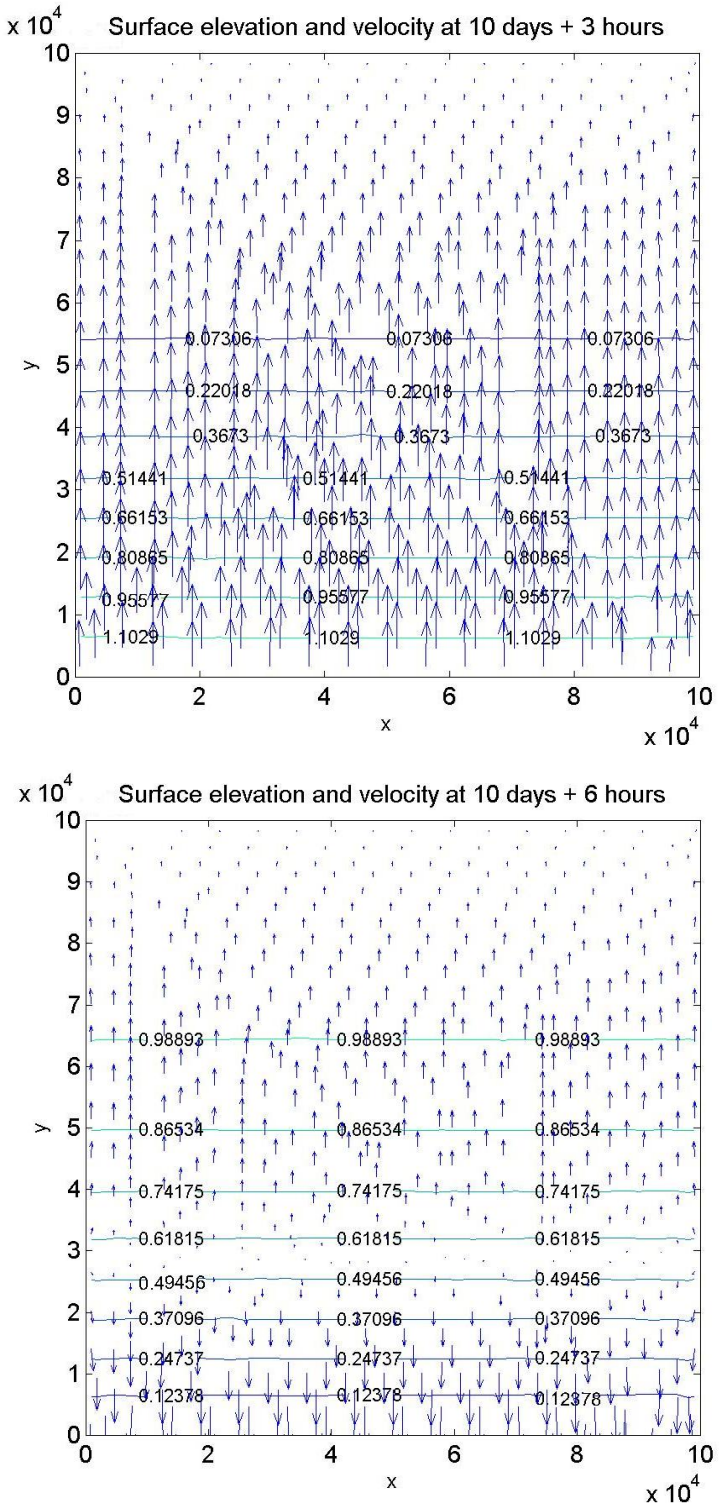


Figure 6.2: Surface elevation contours and velocity profile with no Coriolis force and tidal boundary conditions on a square domain of 2326 elements (3, 6 hours). The flow is quasi-1D, the only force being that of the tide.

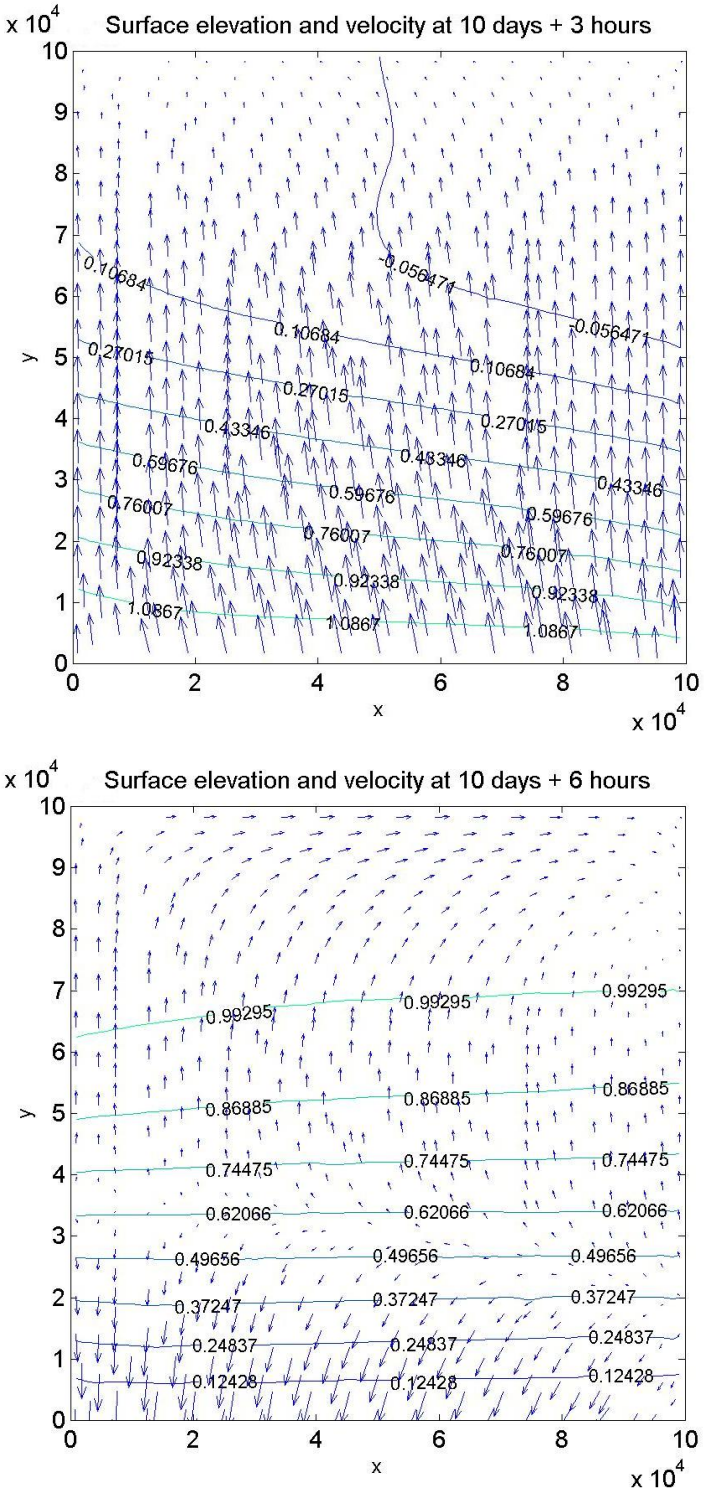


Figure 6.3: a) Surface elevation contours and velocity profile with Coriolis force and tidal boundary conditions on a square domain of 2326 elements (3, 6 hours).

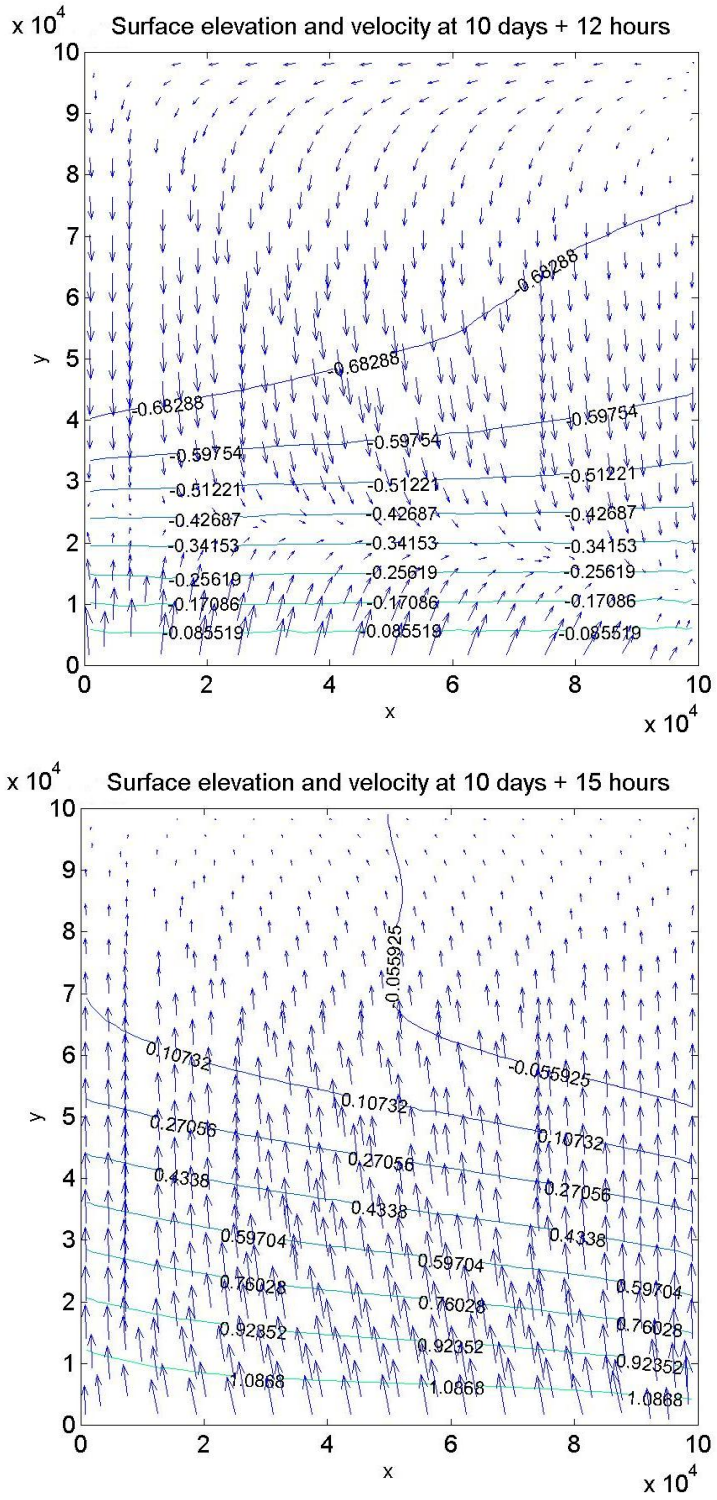


Figure 6.3: b) Surface elevation contours and velocity profile with Coriolis force and tidal boundary conditions on a square domain of 2326 elements (12, 15 hours).

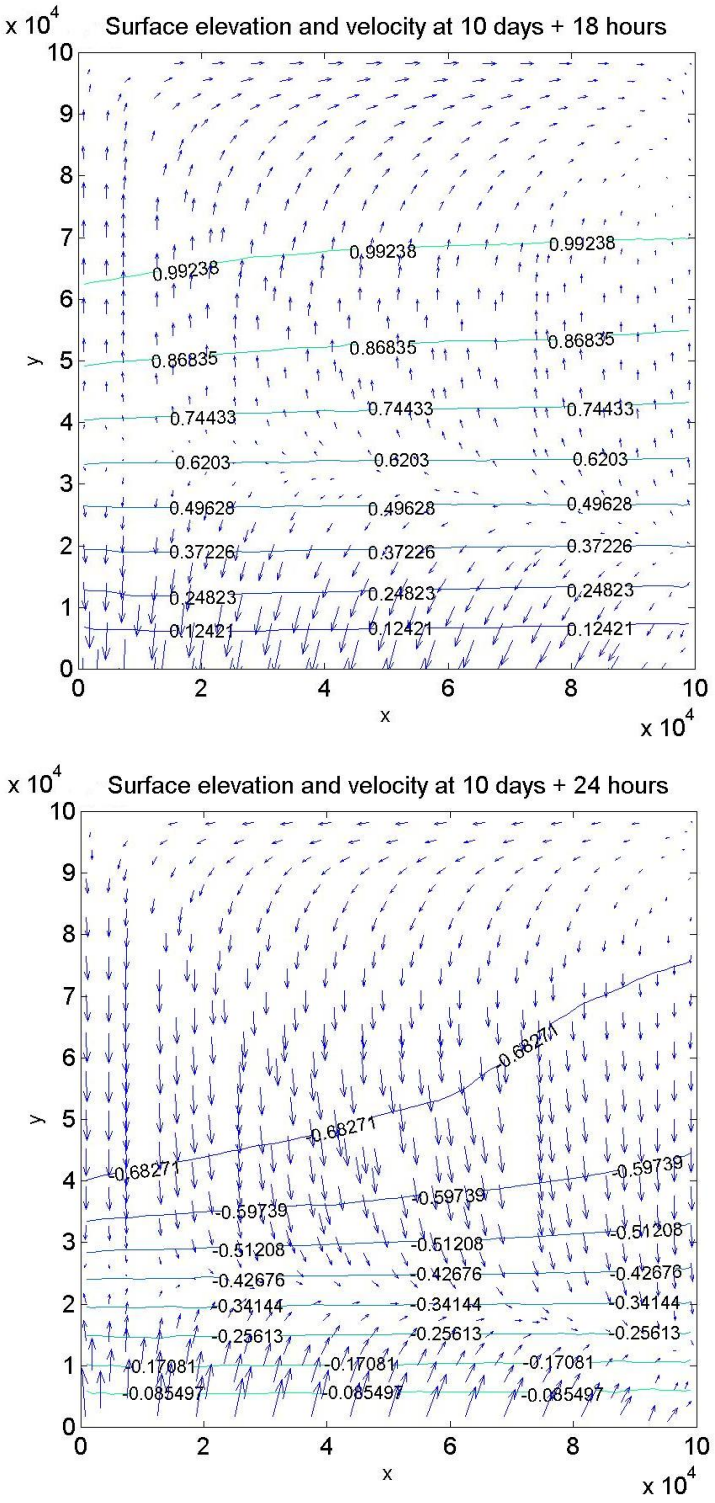


Figure 6.3: c) Surface elevation contours and velocity profile with Coriolis force and tidal boundary conditions on a square domain of 2326 elements (18, 24 hours).

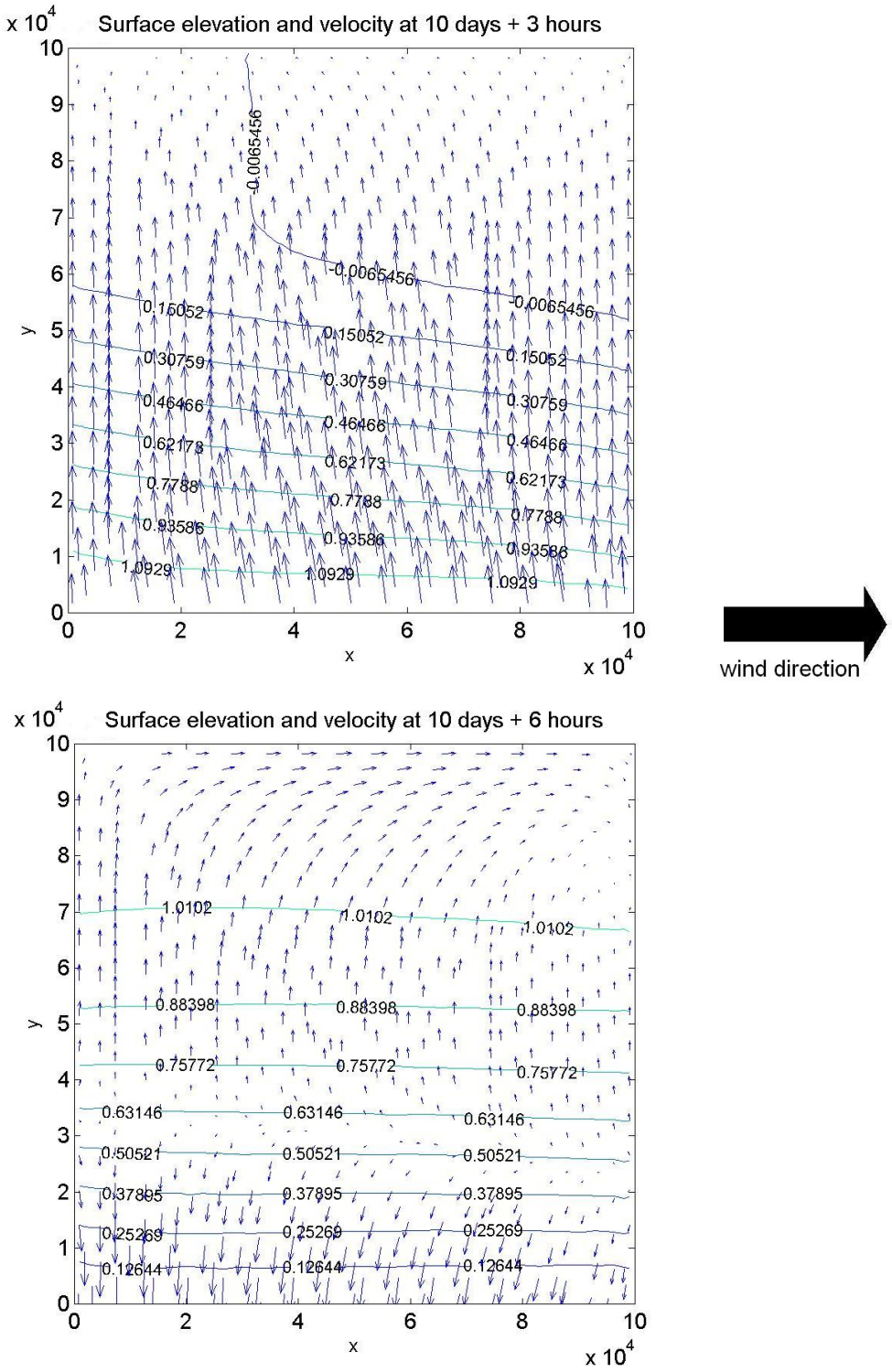


Figure 6.4: a) Surface elevation contours and velocity profile for periodic boundary conditions, westerly wind $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (3, 6 hours).

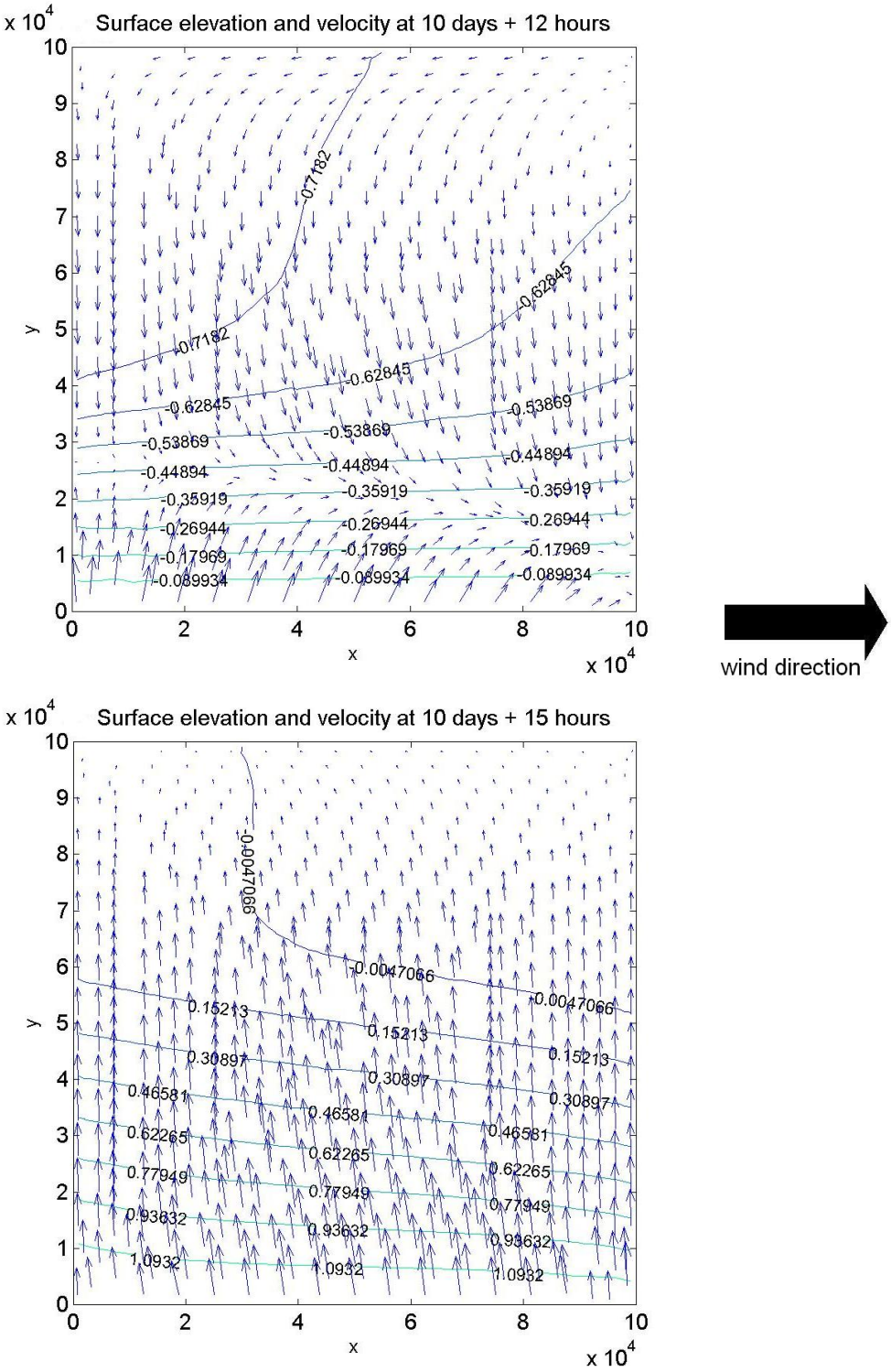


Figure 6.4: b) Surface elevation contours and velocity profile for periodic boundary conditions, westerly wind $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (12, 15 hours).

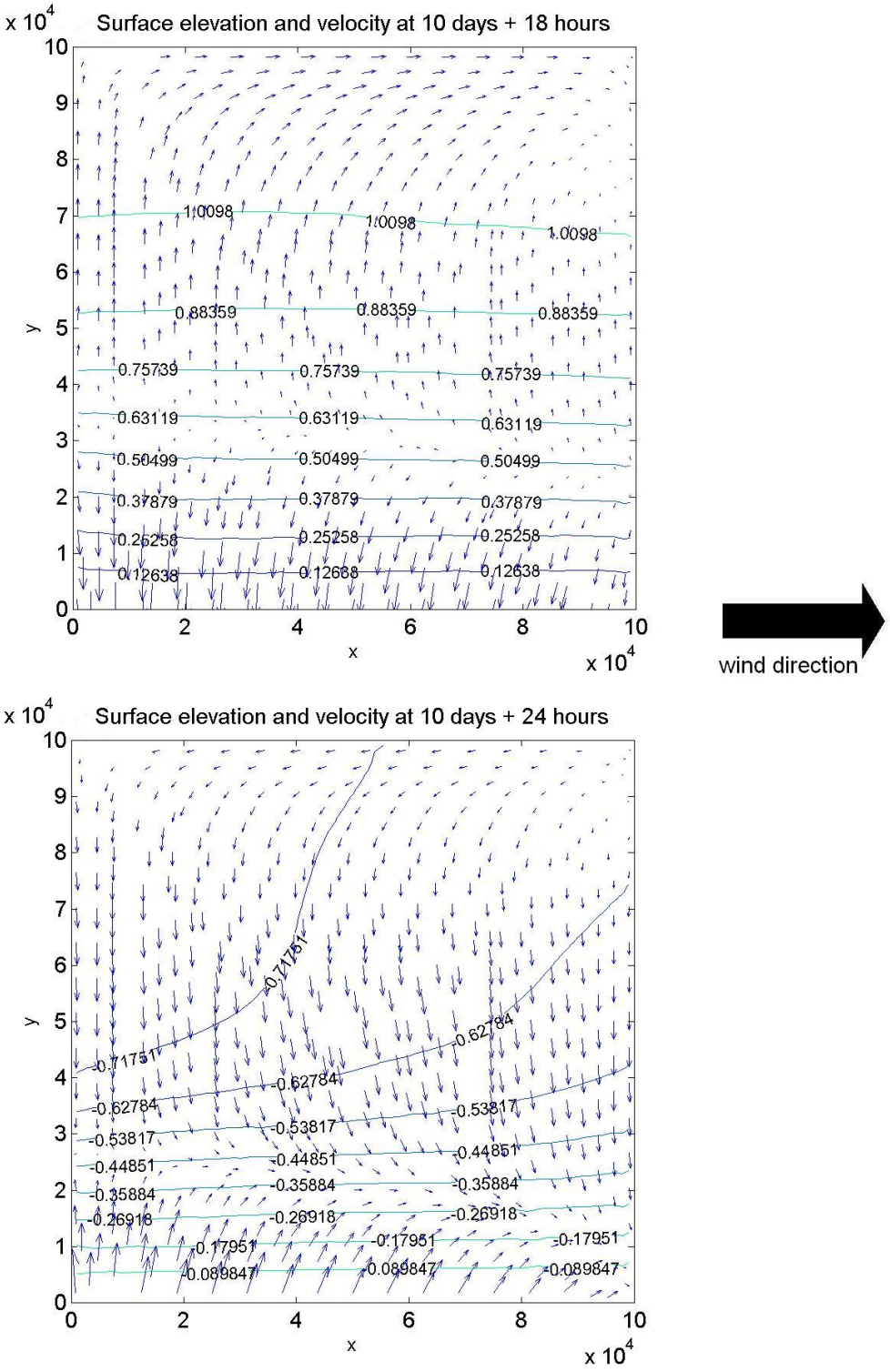


Figure 6.4: c) Surface elevation contours and velocity profile for periodic boundary conditions, westerly wind $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (18, 24 hours).

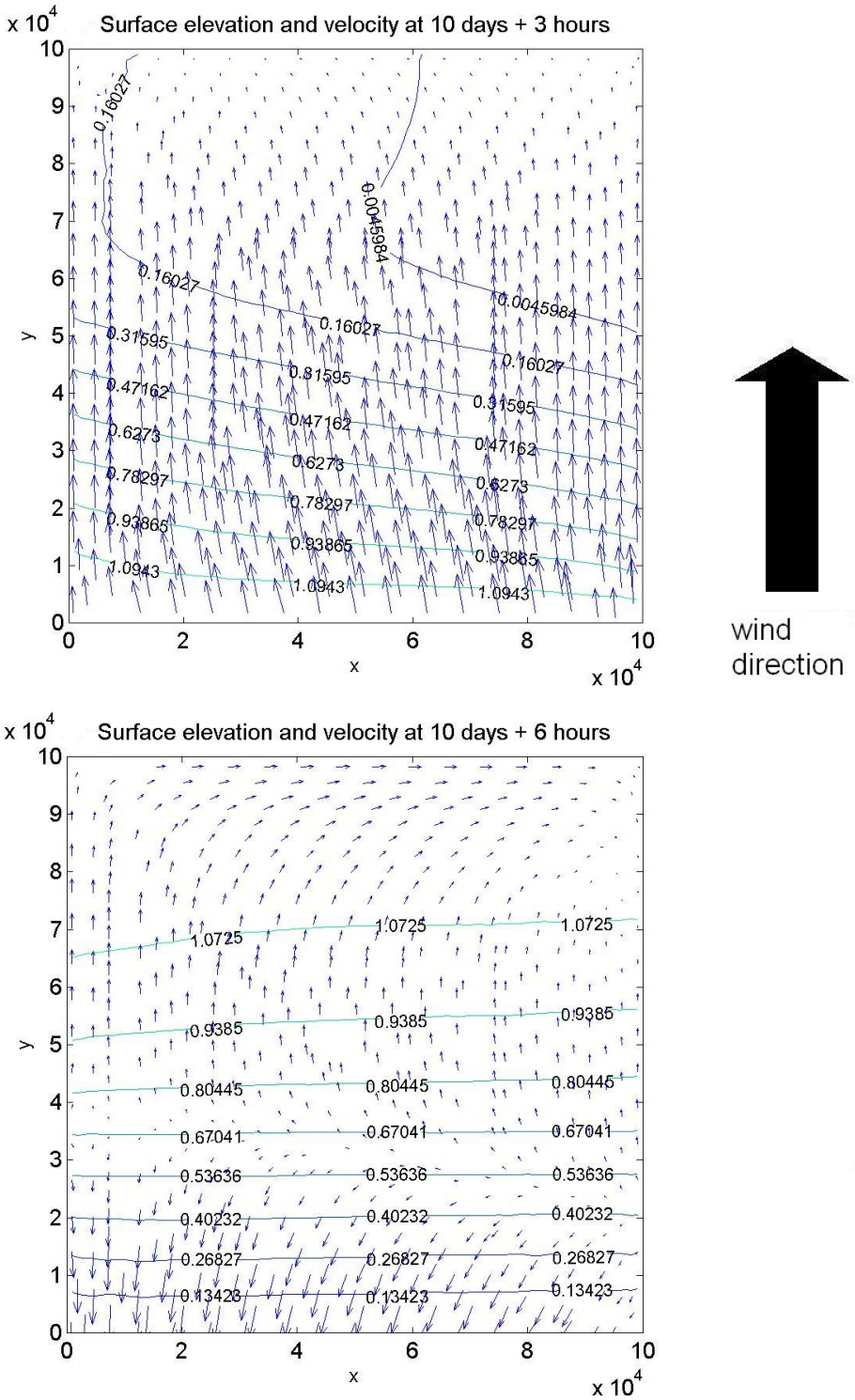


Figure 6.5: a) Surface elevation contours and velocity profile for periodic boundary conditions and southerly wind of $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (3, 6 hours).

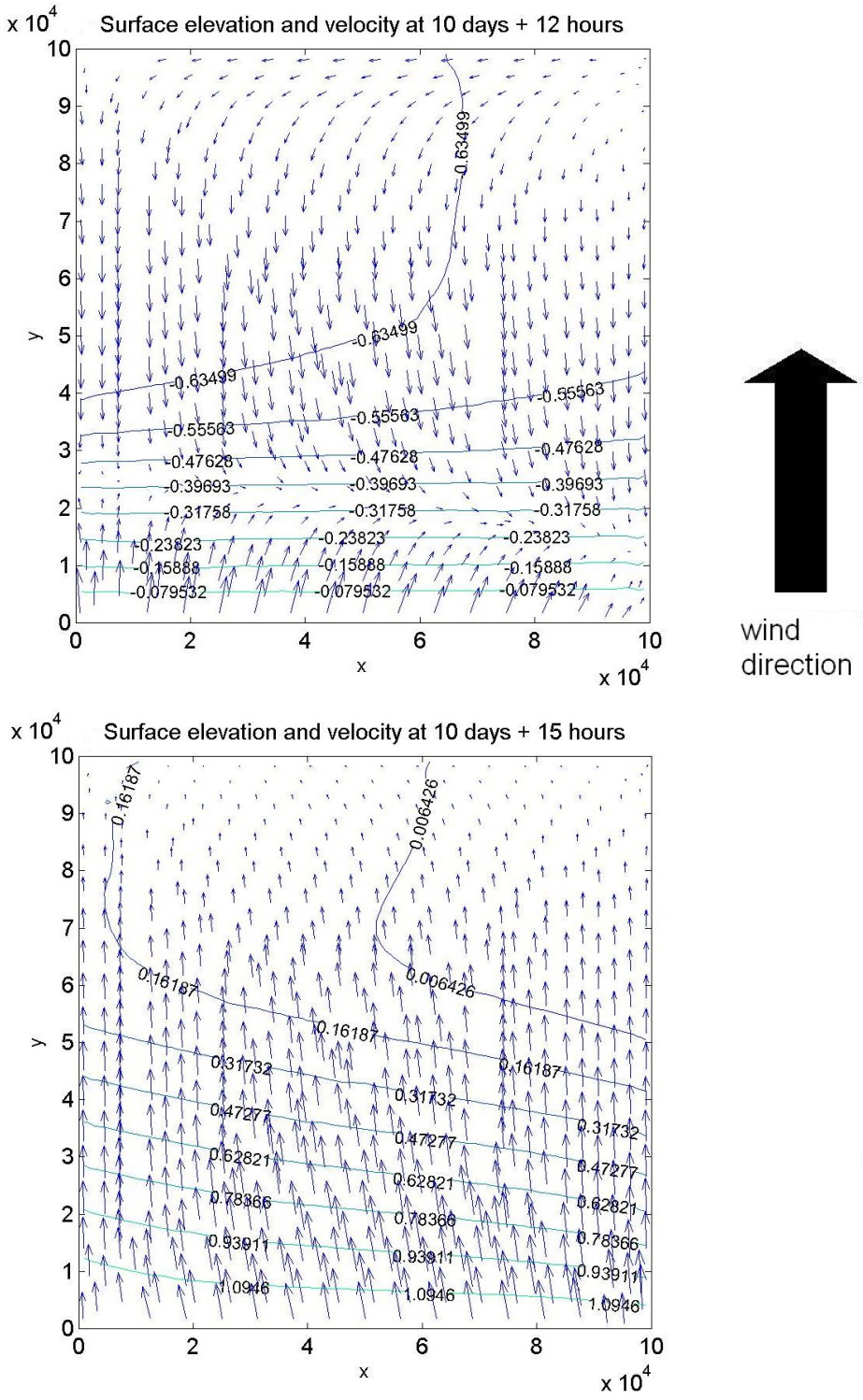


Figure 6.5: b) Surface elevation contours and velocity profile for periodic boundary conditions and southerly wind of $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (12, 15 hours).

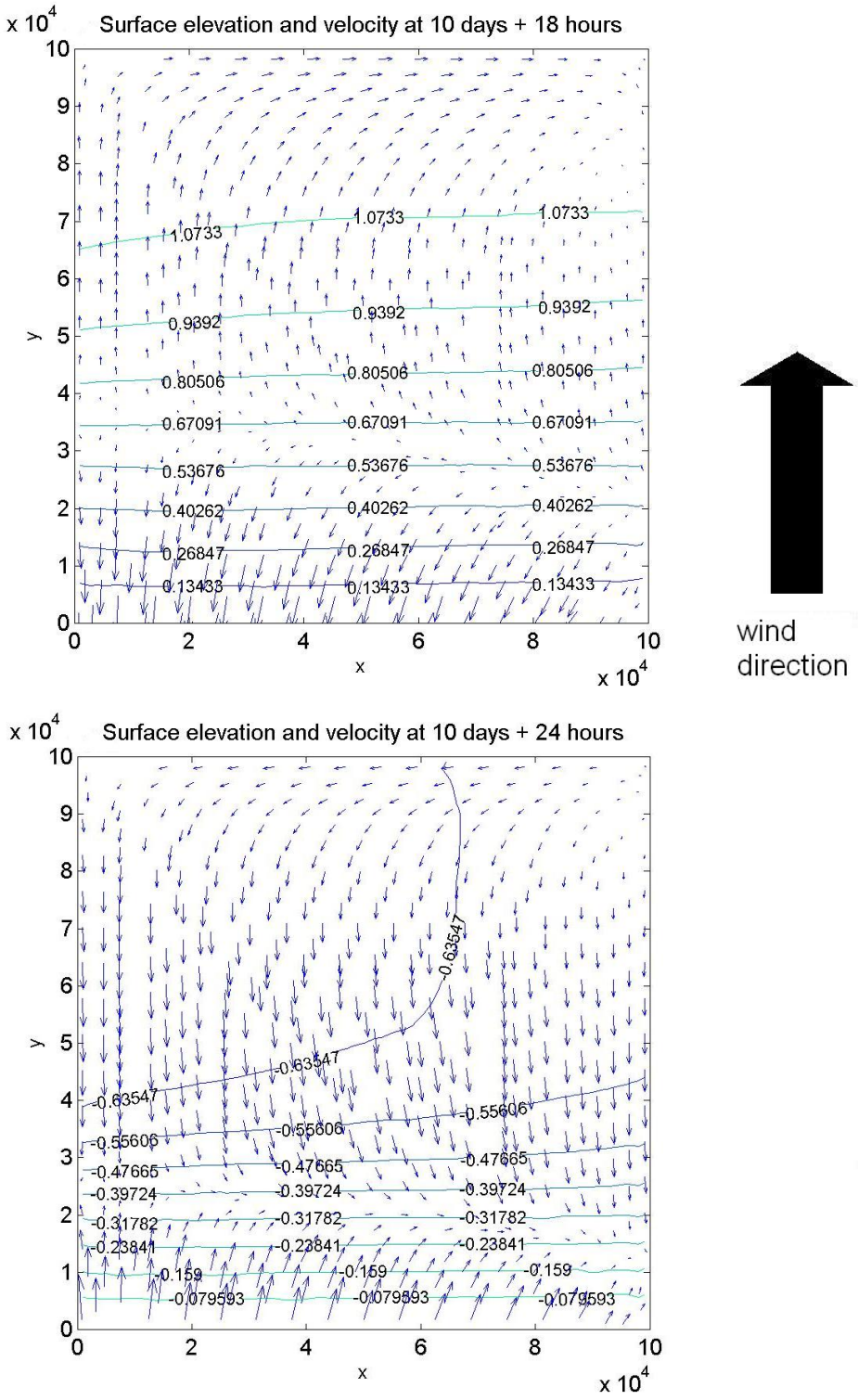


Figure 6.5: c) Surface elevation contours and velocity profile for periodic boundary conditions and southerly wind of $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (18, 24 hours).

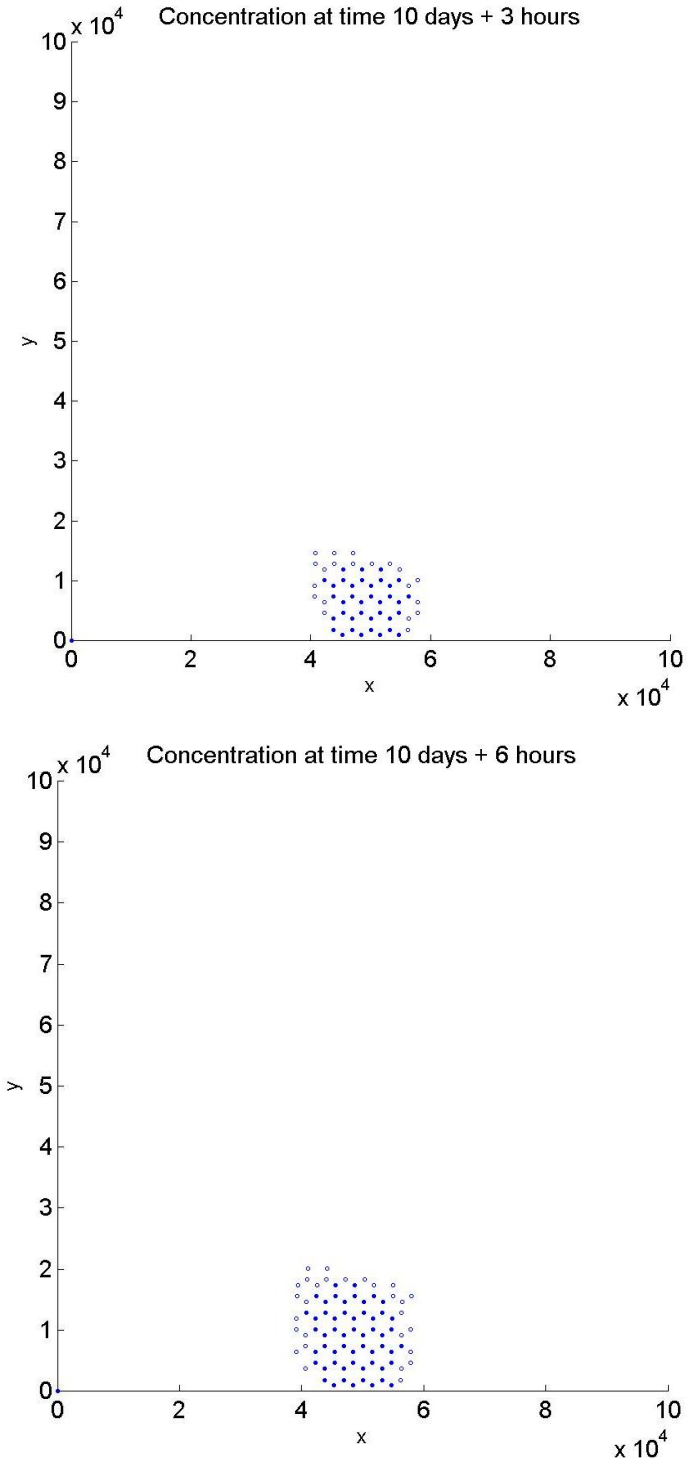


Figure 6.6: a) Scatter of concentration for periodic boundary conditions on a square domain of 2326 elements (3, 6 hours).

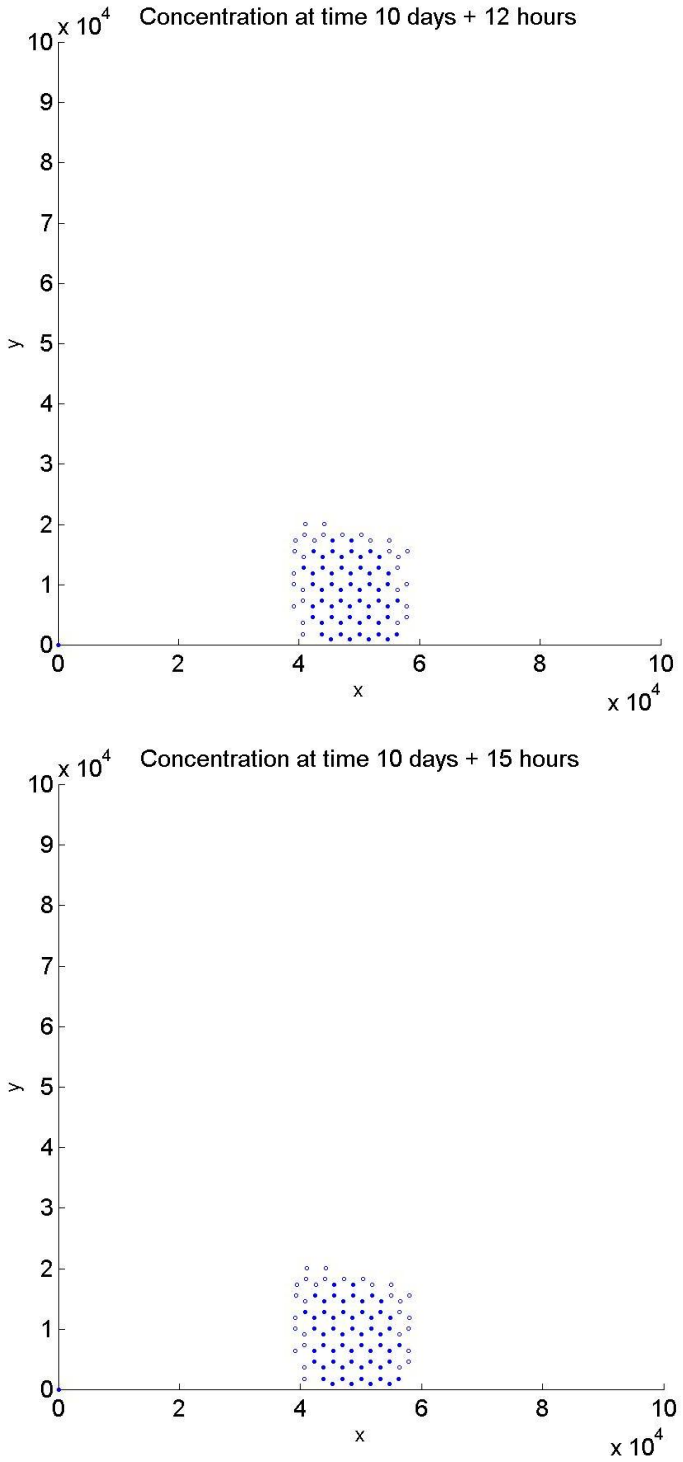


Figure 6.6: b) Scatter of concentration for periodic boundary conditions on a square domain of 2326 elements (12, 15 hours).

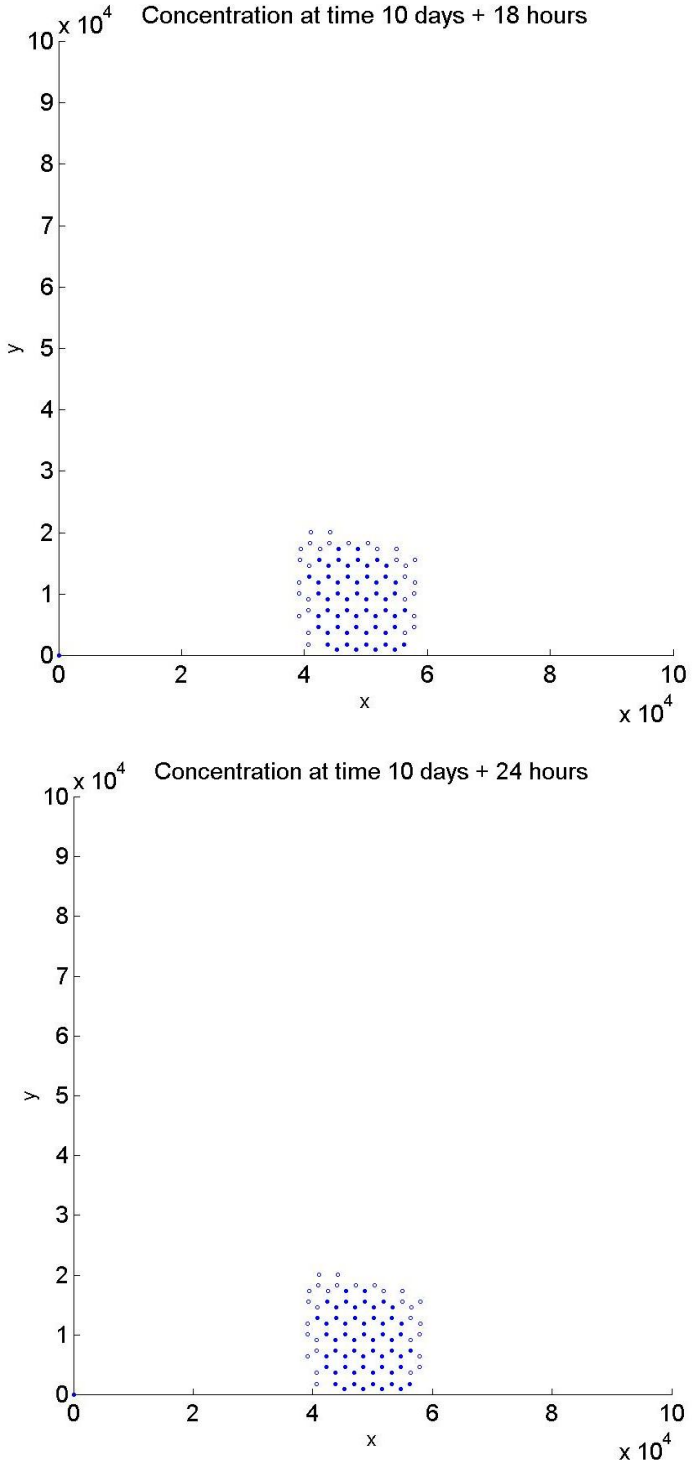


Figure 6.6: c) Scatter of concentration for periodic boundary conditions on a square domain of 2326 elements (18, 24 hours).

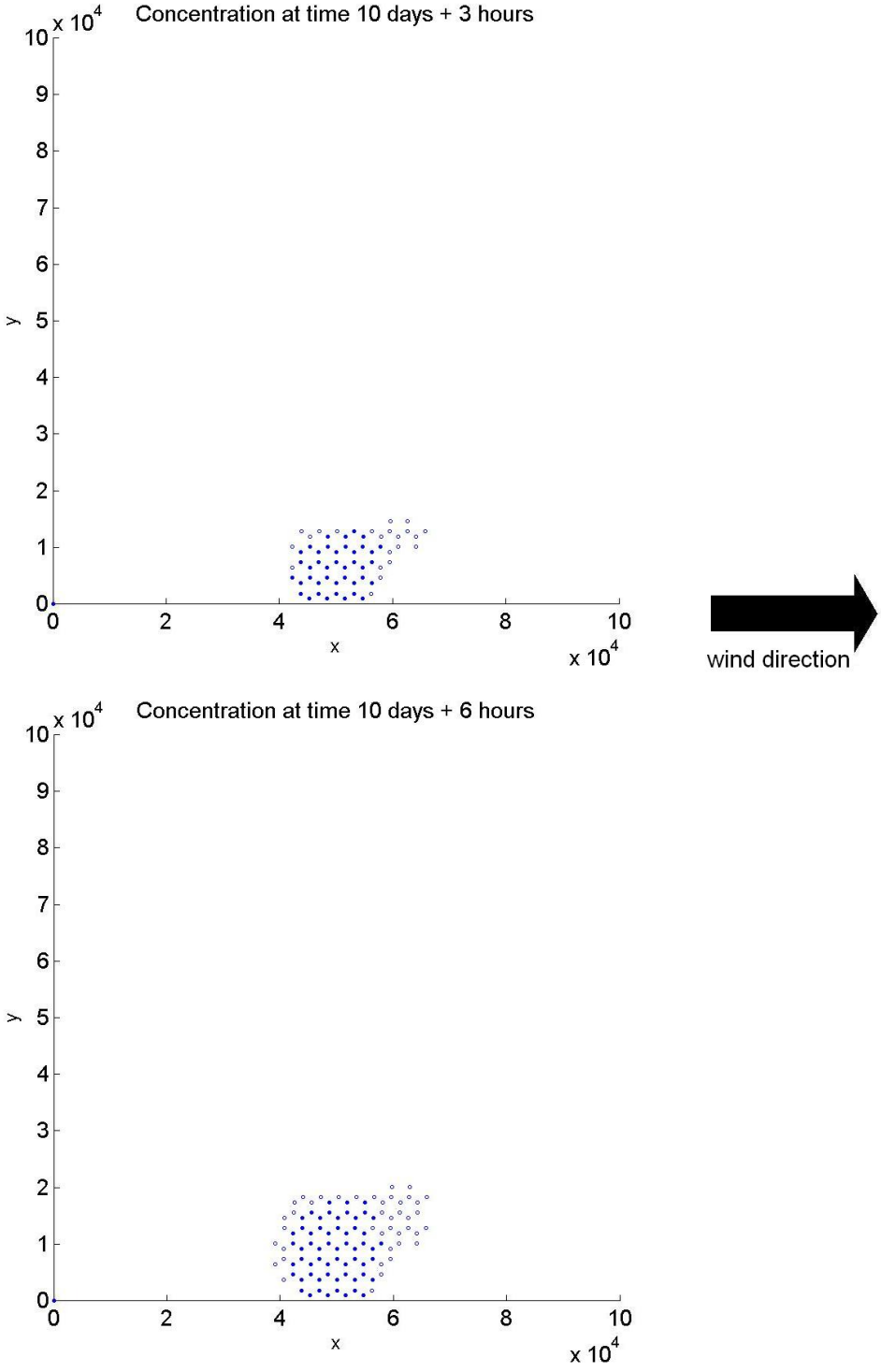


Figure 6.7: a) Scatter of concentration for periodic boundary conditions and westerly wind of $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (3, 6 hours).

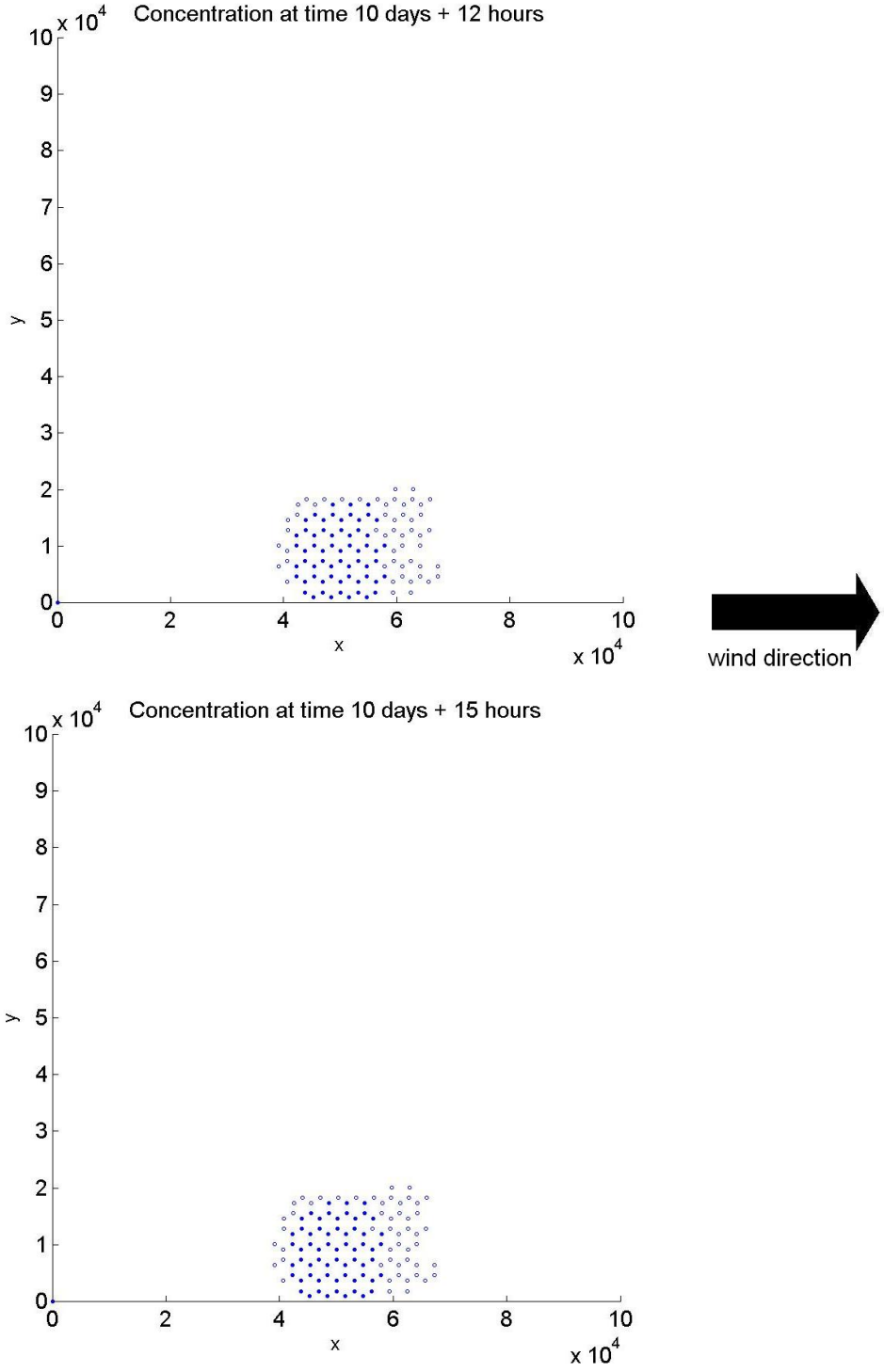


Figure 6.7: b) Scatter of concentration for periodic boundary conditions and westerly wind of $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (12, 15 hours).

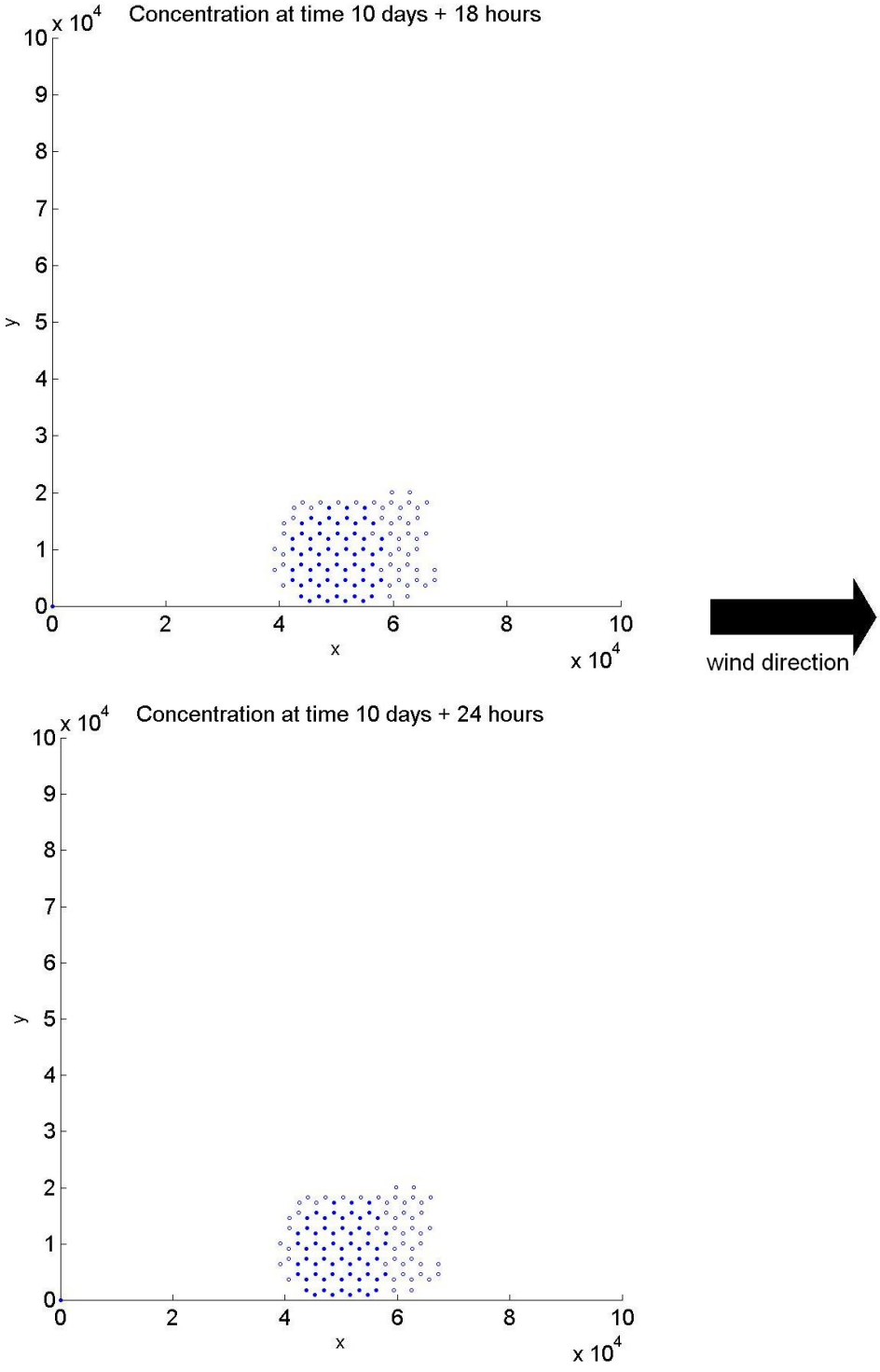


Figure 6.7: c) Scatter of concentration for periodic boundary conditions and westerly wind of $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (18, 24 hours).

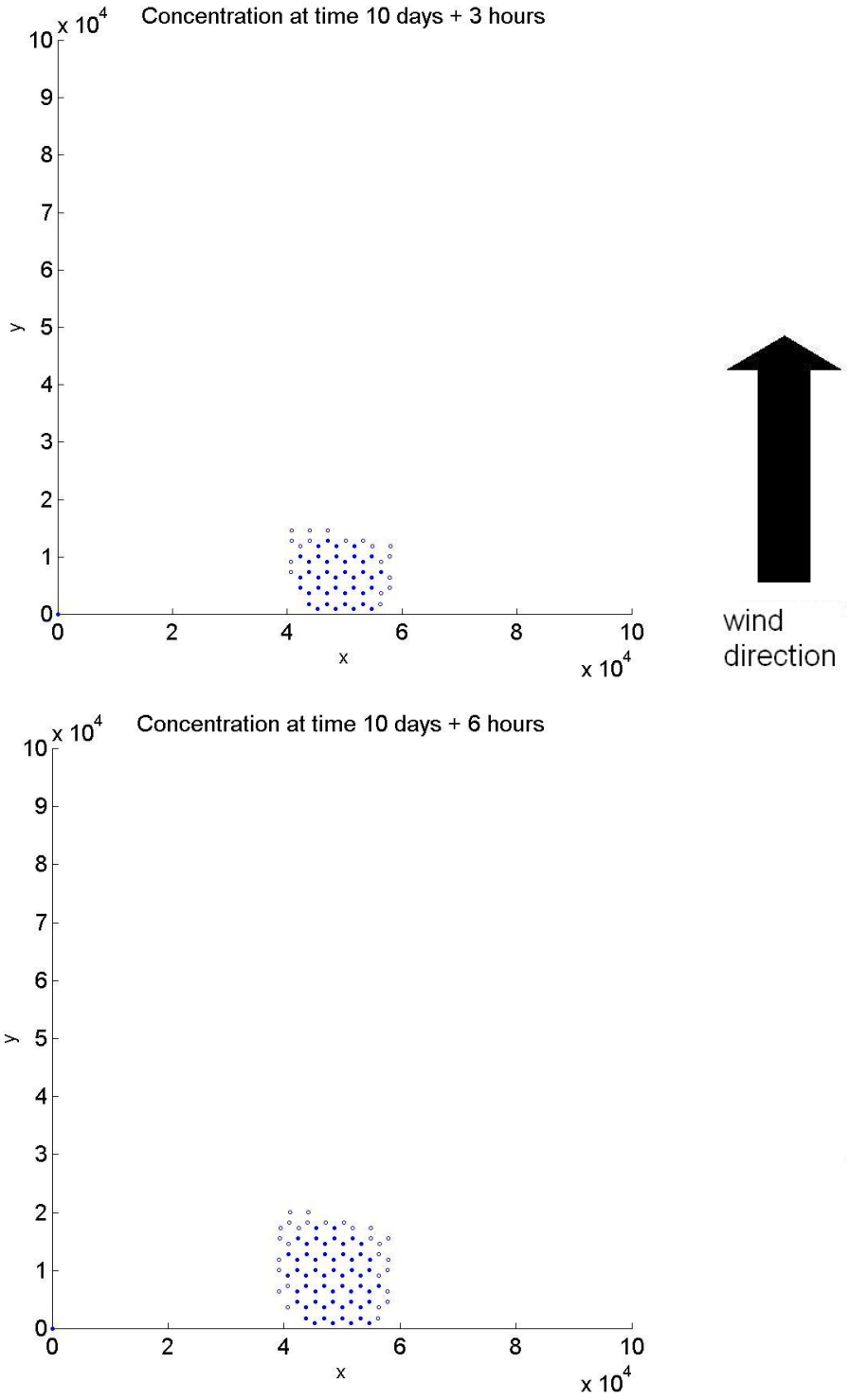


Figure 6.8: a) Scatter of concentration for periodic boundary conditions and southerly wind of $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (3, 6 hours).

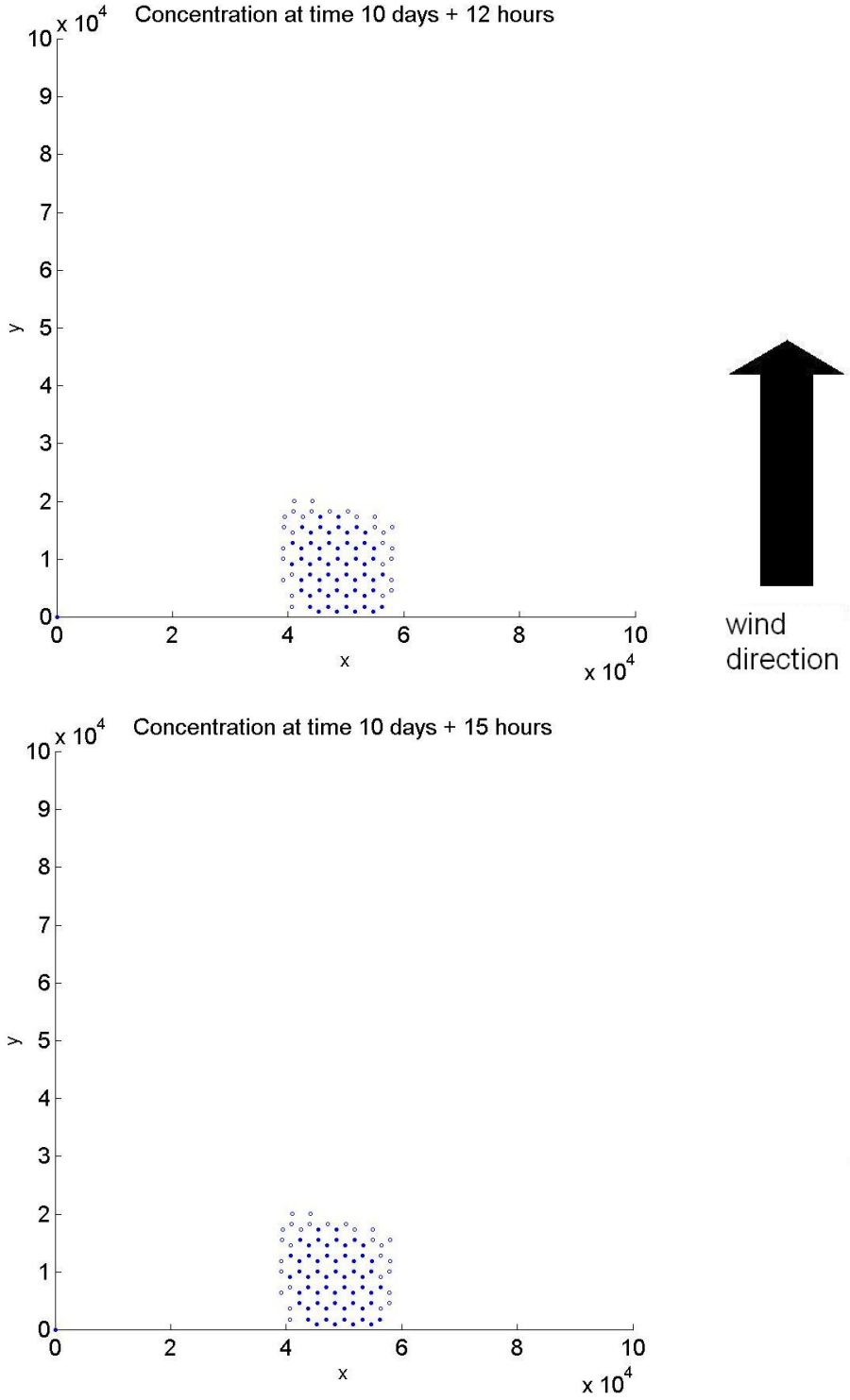


Figure 6.8: b) Scatter of concentration for periodic boundary conditions and southerly wind of $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (12, 15 hours).

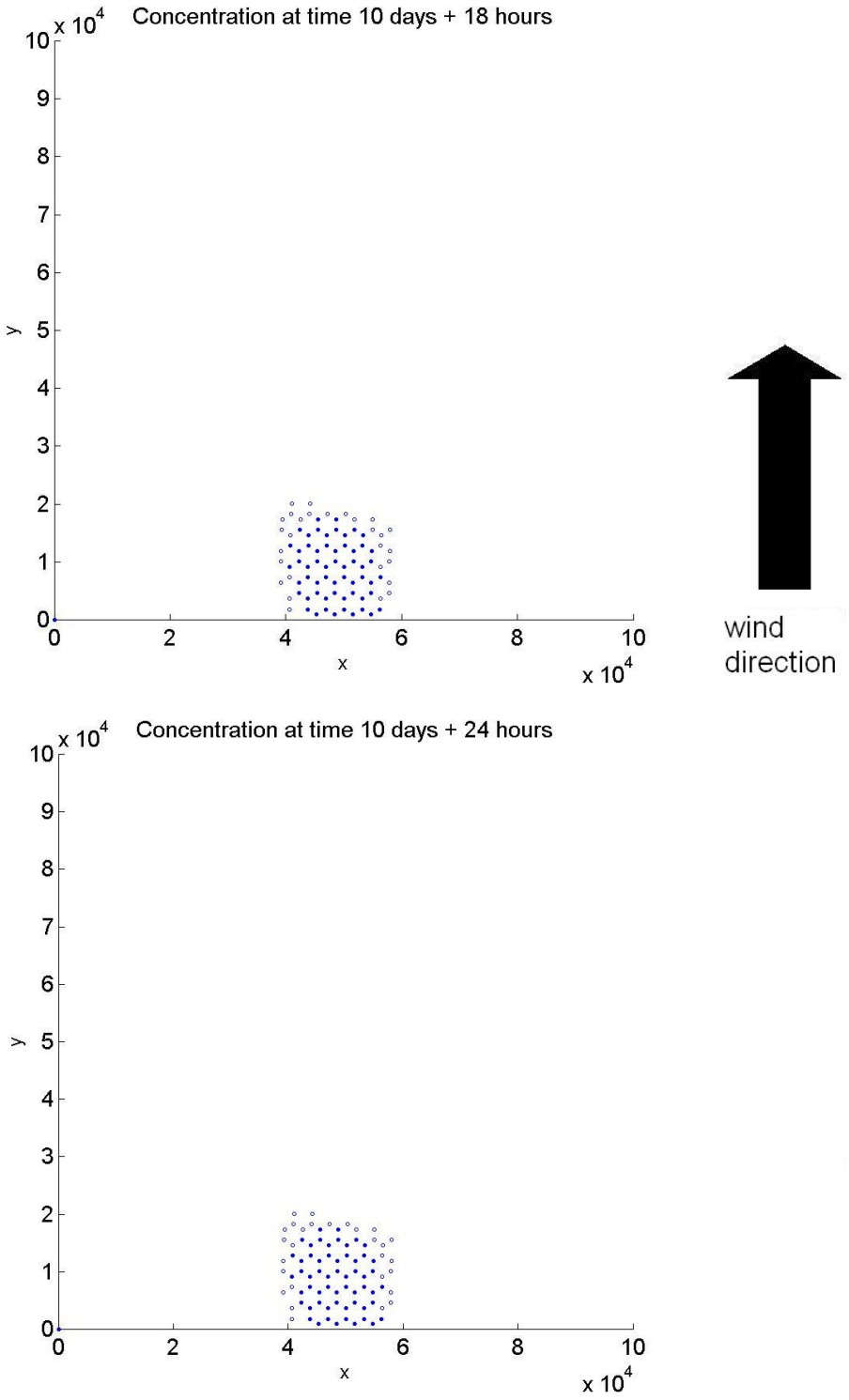


Figure 6.8: c) Scatter of concentration for periodic boundary conditions and southerly wind of $5\,\mathrm{ms}^{-1}$ on a square domain of 2326 elements (18, 24 hours).

Figure 6.9: Satellite image of Port Phillip bay, Melbourne. (From NASA's globe software, via `commons.wikimedia.org/wiki/Image:Port_Phillip_Bay.png`.) The narrow entrance to the Port can be seen in the southwest part. The city of Melbourne is on the northeast side of the bay.

6.2 Time-dependent boundary conditions (circular basin)

The shape of Port Phillip bay in Melbourne is approximately circular, about 90 km across and 10 m deep at its deepest point, with one single narrow inlet at the south west (see Figure 6.9). During mid tides, very strong currents are generated in this narrow passage. According to Black *et al.* (1993), velocities can reach $3\,\mathrm{ms}^{-1}$ in the bay's entrance.

In this section we use the circular basin-like domain from Chapter 5 with the same type of ocean boundary as that of Port Phillip bay, *i.e.* a narrow gap on the south side for water to enter and leave with the tide. The domain was modified to make it more realistic, by creating deeper bathymetry near the ocean boundary so that the entrance was around a metre deep.

The domain used was a circular basin of mesh density of 2484 elements, radius $r_0 = 45\,\mathrm{km}$, maximum depth $b_0 = -10\,\mathrm{m}$ (see (5.9)). A small gap of six elements on the south edge of the domain was chosen to represent the ocean boundary of the bay. Here boundary conditions were set as periodic to represent the tidal effects, in the same way as for the square. All

the other boundary conditions were reflective. The test was run for a simulation time of 10 days before any data were recorded, as before, with fixed timestep of size dt $= 1\,\mathrm{s}$, Manning's $n = 0.03\,\mathrm{m}^{-1/3}\mathrm{s}$ (Larsen (2007), Anderson (2002)), and minDepth $1 \times 10^{-2}\,\mathrm{m}$. The larger value for minimum depth is to ensure no instabilities develop in the flow as $h \to 0$. The Courant number was around 0.006, and the total time taken for the runs was about 10 hours 30 minutes for all the circular basin tests.

6.2.1 Comparing Coriolis force

The effect that Coriolis force has on the symmetry of the flow can be seen in surface elevation and velocity profiles given in Figures 6.10 (no Coriolis) and 6.11a)-c) (with Coriolis parameter $-8.7 \times 10^{-5}\,\mathrm{s}^{-1}$, corresponding to 37°S, the latitude of Melbourne), which show the direction and magnitude of the flow, scaled by a factor of 4, with average values given in Table 6.3. Without Coriolis it is clear from the velocity vectors and the contour lines that the surface elevation and flow are symmetric; the anti-clockwise spin of outflowing water can be seen at 6 hours with Coriolis. Arrows at 3 hours show that the flow is directed west with Coriolis force, and this means deeper water on the west of the bay.

The periodic nature of the boundary condition can, again, be verified by noticing the similarity of both plots and average velocities at pairs 3 and 15 hours, 6 and 18 hours, and 12 and 24 hours (Table 6.3 and Figures 6.11a)-c)).

Note the very high velocities at high tides (3, 15 hours) near the ocean boundary. This is due to the very shallow water here, of about one metre, and agrees with what is known about Port Phillip tides (Black *et al.* (1993)). Because the volume at the ocean boundary is relatively small, the flow is forced through faster.

Although the effect of Coriolis force can be clearly seen, it is not as strong as in the case of the square bay. This is because areas of high friction (shallow bathymetry) and the fast flow at the ocean boundary are as important as the Coriolis force, which is not the case for the square bay where bed friction effects are small.

time (hours) + 10 days	$(\bar{u}, \bar{v})$	time (hours) + 10 days	$(\bar{u}, \bar{v})$
3	-0.0225, 0.0328	15	-0.0227, 0.0329
6	0.0253, 0.0056	18	0.0256, 0.0051
12	0.0200, -0.0064	24	0.0203, -0.0067

Table 6.3: Average velocities in the u and v directions relating to Figures 6.11a)-c). The pairs (3, 15), (6, 18), and (12, 24) hours have identical velocities correct to the third decimal place.

6.2.2 Surface elevation and wind

In two further tests, wind of $5\,\mathrm{ms}^{-1}$ was imposed. In the first case, the wind direction was westerly, in the second, southerly, to compare with the results from the square.

Results of the surface elevation and velocity profile with wind are shown in Figures 6.12a)-c) (westerly wind) and 6.13a)-c) (southerly wind). Compare with the plots with no wind in Figures 6.11a)-c). The velocity profiles show a reproduction of the analytical result from Chapter 5, where wind generates two rotating gyres. In these cases, the effects of

- tide can be seen as the gyres vary in intensity throughout the tidal cycle,
- Coriolis can be seen as the divide between the two gyres is neither as straight or exactly in the x or y plane as it was in the test of Chapter 5.

The surface elevation is very much affected by the wind for the circular basin, unlike for the square, where flow is quasi-1D. Larsen (2007) found that areas of higher bed friction limit tidal effects. So because of the basin's varying bathymetry, the strength of influence of the wind becomes more important than that of the tide.

These tests show that flow in the circular basin is more susceptible to the force of wind than flow in the square. This is because there are areas of shallow water that are more subject to shear stress from the wind and they therefore have a greater impact on the pattern of flow.

6.2.3 Effect of wind direction on pollutants

The central four elements of the ocean boundary were also given initial concentration of nominal value $0.3\,\mathrm{kg\ m^{-3}}$. The boundary condition for this contaminant was set as before so that at each timestep the relevant ghost elements were set to be $hc_{ghost} = -0.3b$, and updated at each timestep.

Passive tracer

Figures 6.14a)-c) show scatter plots for a contaminant released at the entrance to the bay. The empty points indicate elements with a concentration, c, of $0.01 < c < 0.05\,\mathrm{kg\ m^{-3}}$ and the filled points represent elements with $c > 0.05\,\mathrm{kg\ m^{-3}}$.

The surface elevation profiles in Figures 6.11a)-c) show the pattern of flow as fan-like, and the contaminant follows this pattern. We also note that flow in the east of the domain is slower and surface elevation lower due to Coriolis, which means that contaminant is not spread out so quickly in the east. It appears to have reached a steady state as its position or pattern does not change significantly over the 24 hour period shown.

Passive tracer with wind

The same simulation was run twice more, once with a westerly wind and once with a southerly wind, of strength $5\,\mathrm{ms^{-1}}$. Results of the tracer are shown in Figures 6.15a)-c) (westerly wind) and 6.16a)-c) (southerly wind).

Under the westerly wind, the contaminant follows the path of the more southerly gyre but only occupies the eastern part of the bay due to the wind pushing it to that side.

In the case of southerly wind, contaminant spreads out to both sides of the ocean boundary where there are areas of high flow velocity. Contaminant spreads slightly more to the east than the west; this is accounted for by Figure 6.13a)-c) which shows the surface elevation to be slightly higher on the east side.

In both cases, the contaminant stays in an area subject to high shear, and is easily moved by the wind. The distribution is more striking than in the case of a square, as the effect of shear is no longer uniform throughout the domain.

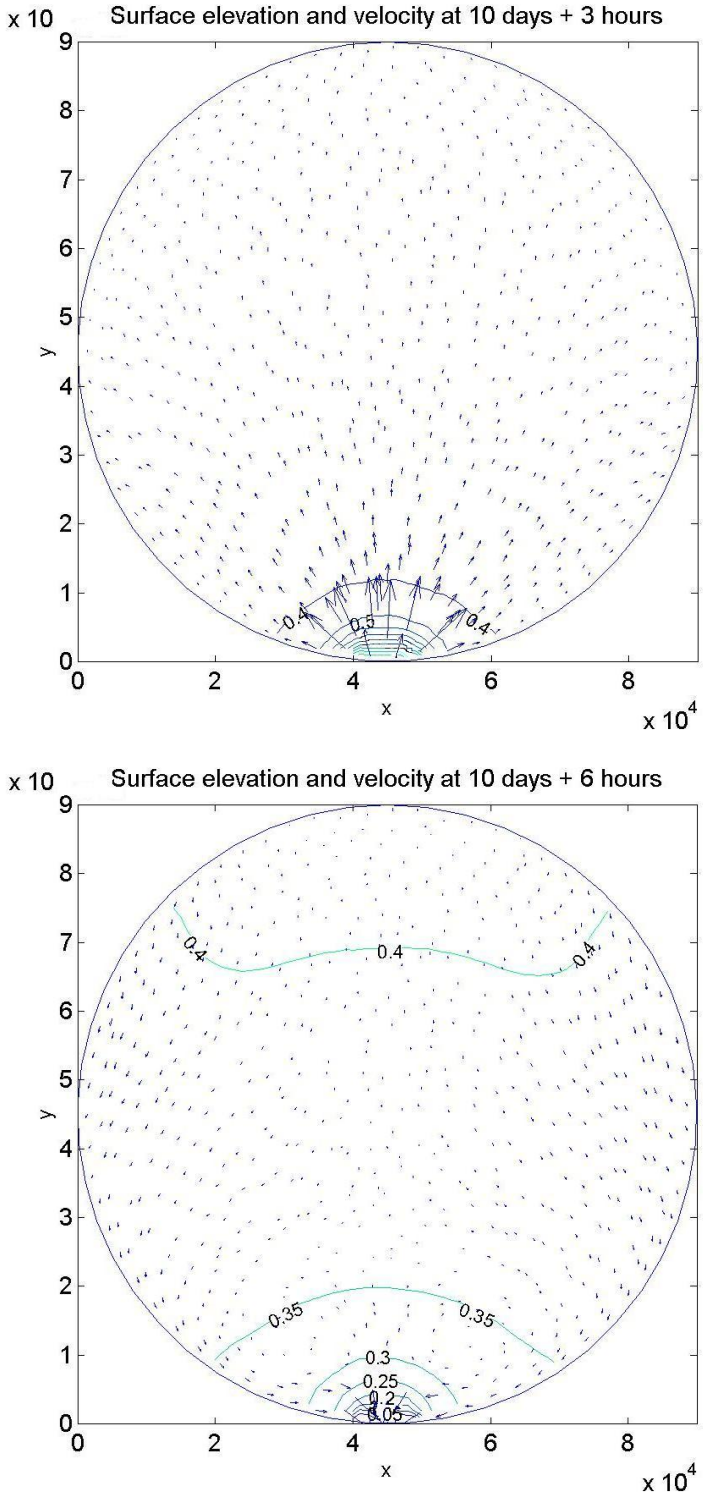


Figure 6.10: Surface elevation contours and velocity profile for periodic boundary conditions with no Coriolis force on a circular domain of 2484 elements (3, 6 hours). The flow is symmetric.

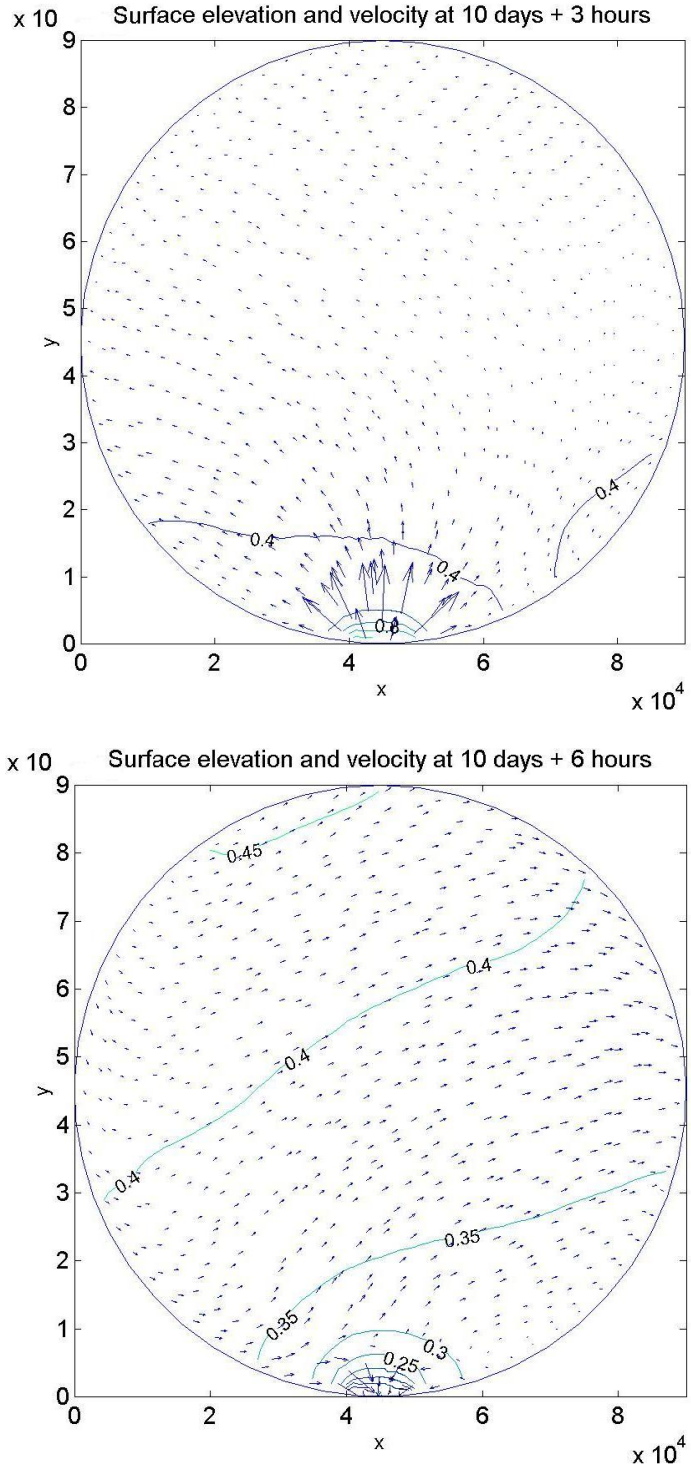


Figure 6.11: a) Surface elevation contours and velocity profile for periodic boundary conditions with Coriolis force on a circular domain of 2484 elements (3, 6 hours).

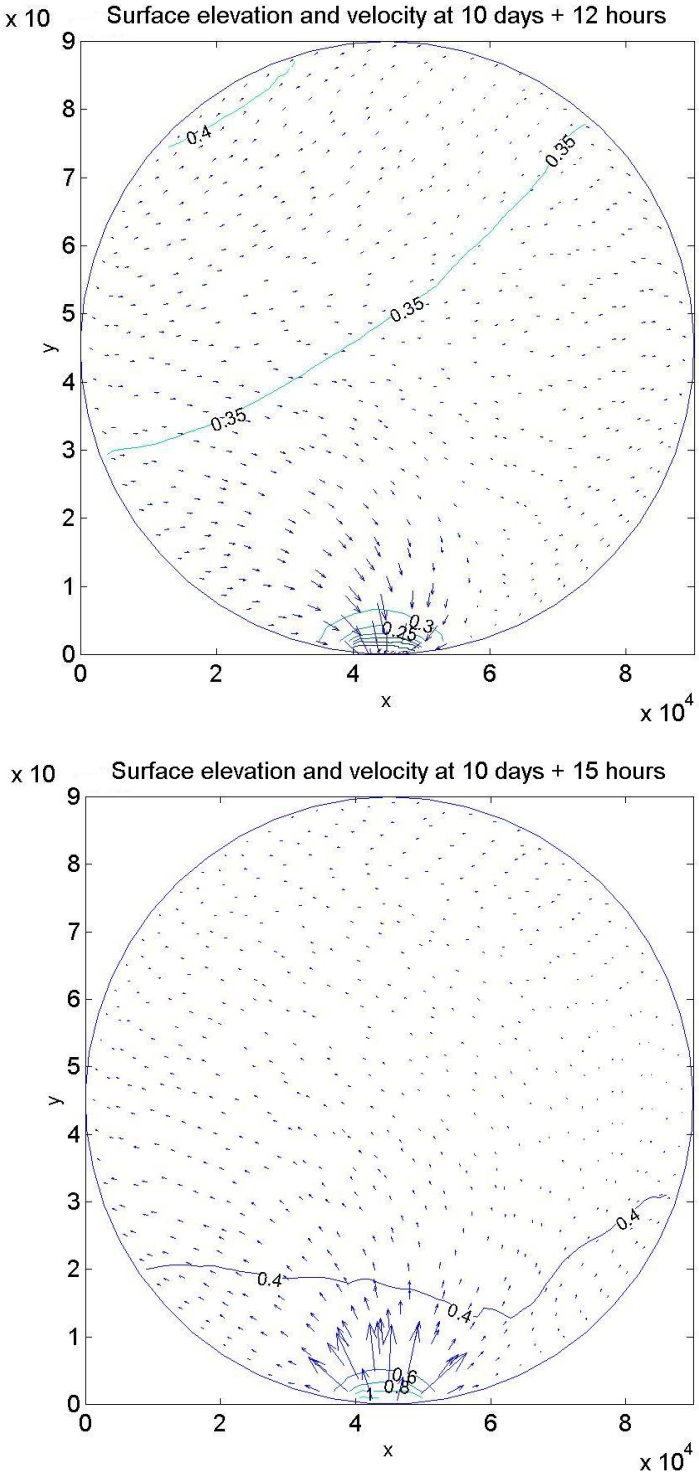


Figure 6.11: b) Surface elevation contours and velocity profile for periodic boundary conditions with Coriolis force on a circular domain of 2484 elements (12, 15 hours).

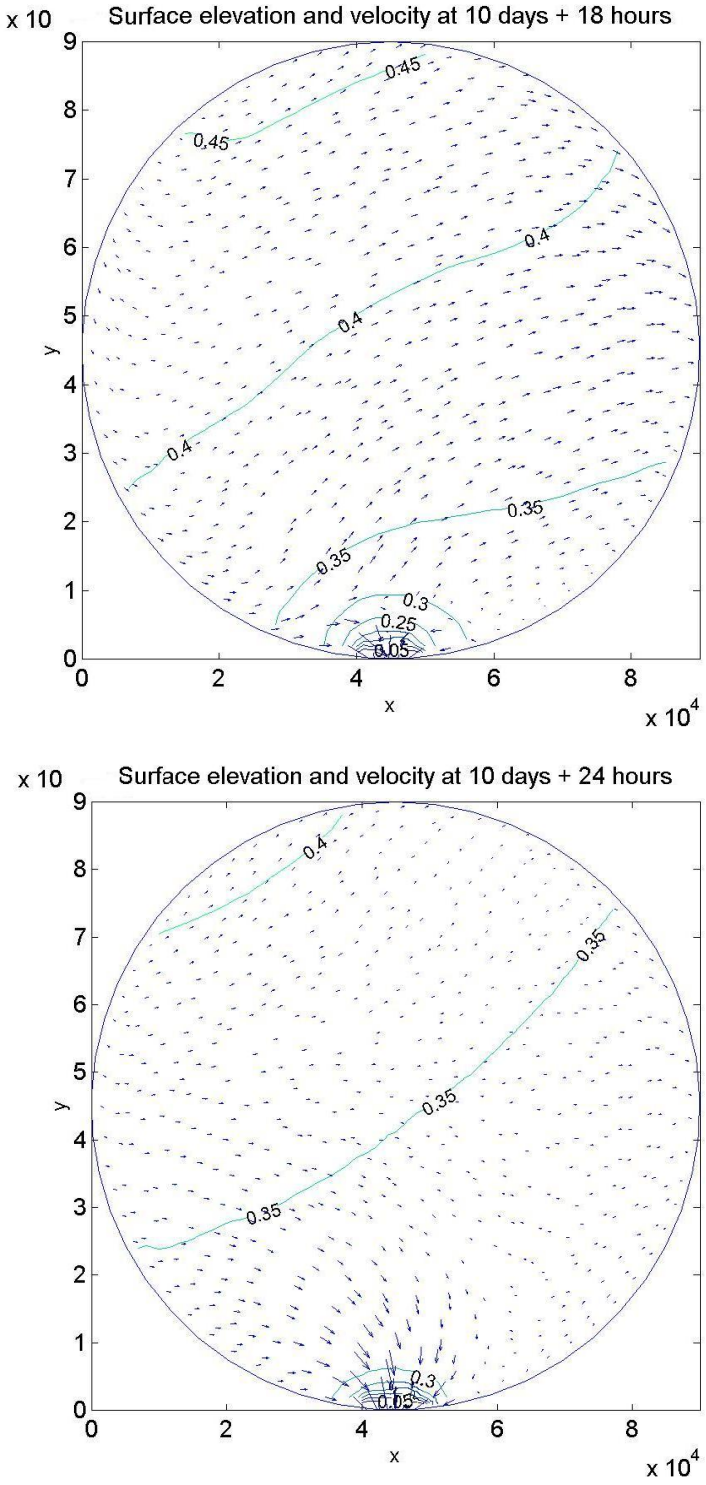


Figure 6.11: c) Surface elevation contours and velocity profile for periodic boundary conditions with Coriolis force on a circular domain of 2484 elements (18, 24 hours).

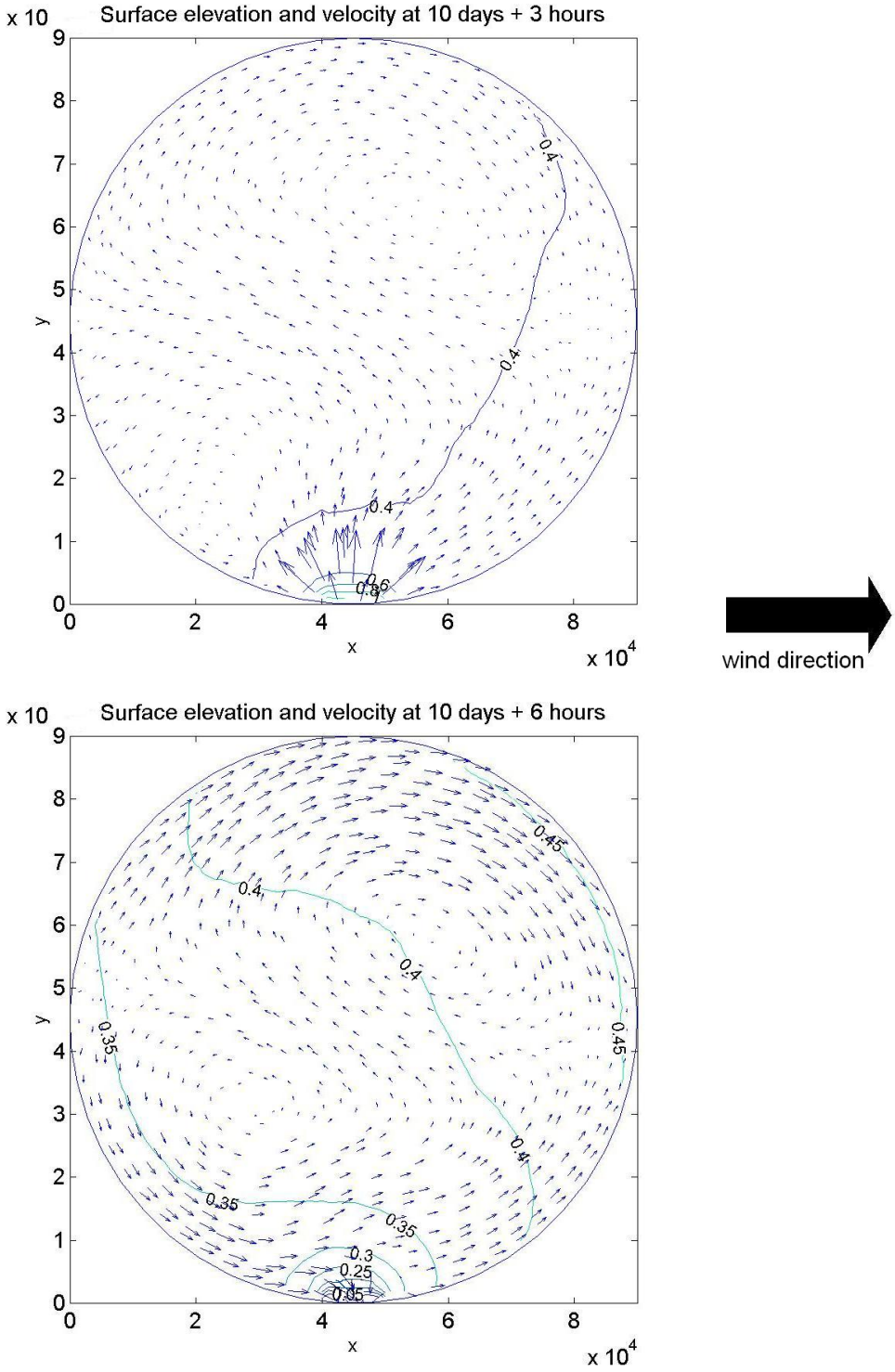


Figure 6.12: a) Surface elevation contours and velocity profile for periodic boundary conditions and a westerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (3, 6 hours).

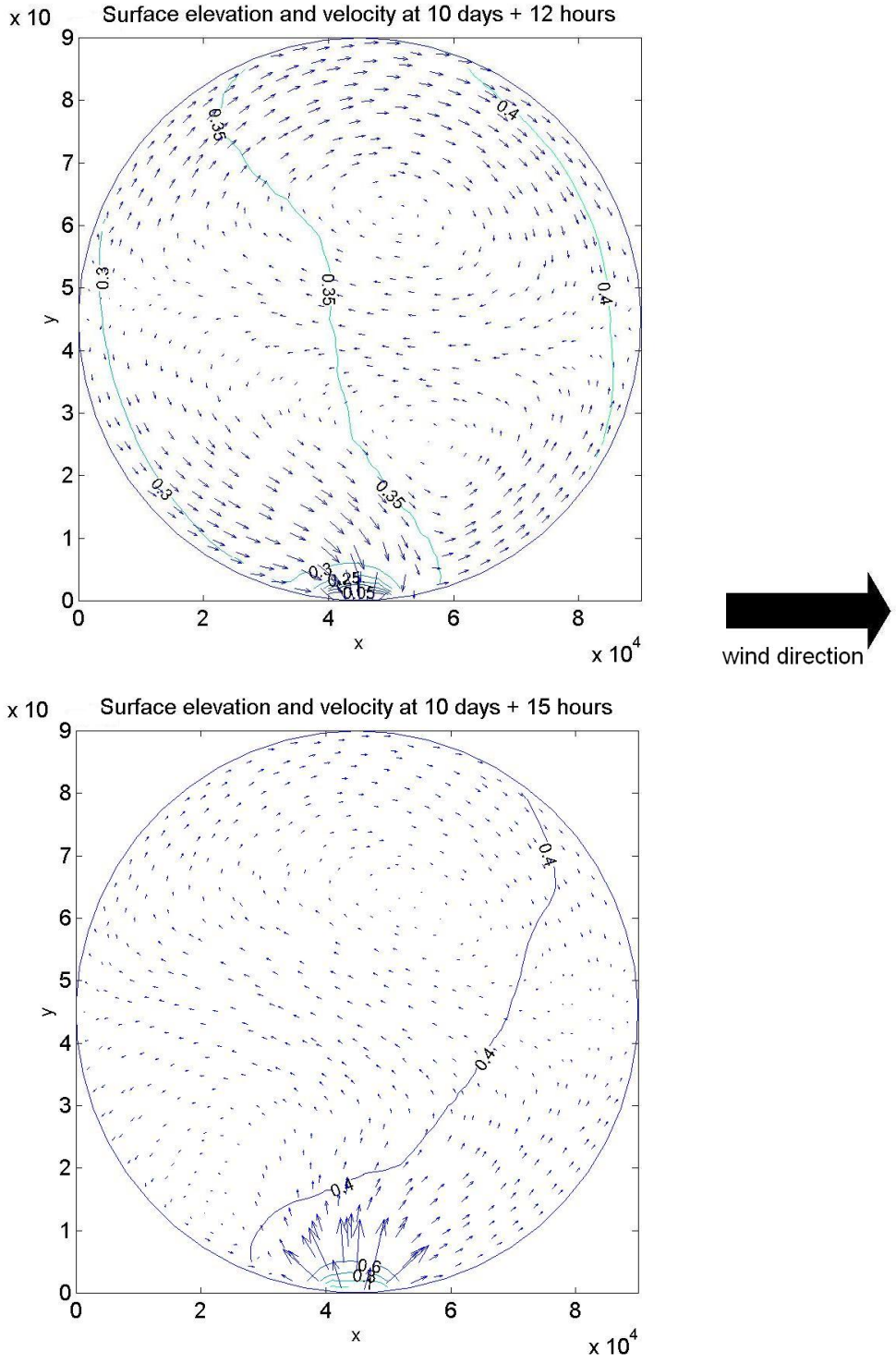


Figure 6.12: b) Surface elevation contours and velocity profile for periodic boundary conditions and a westerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (12, 15 hours).

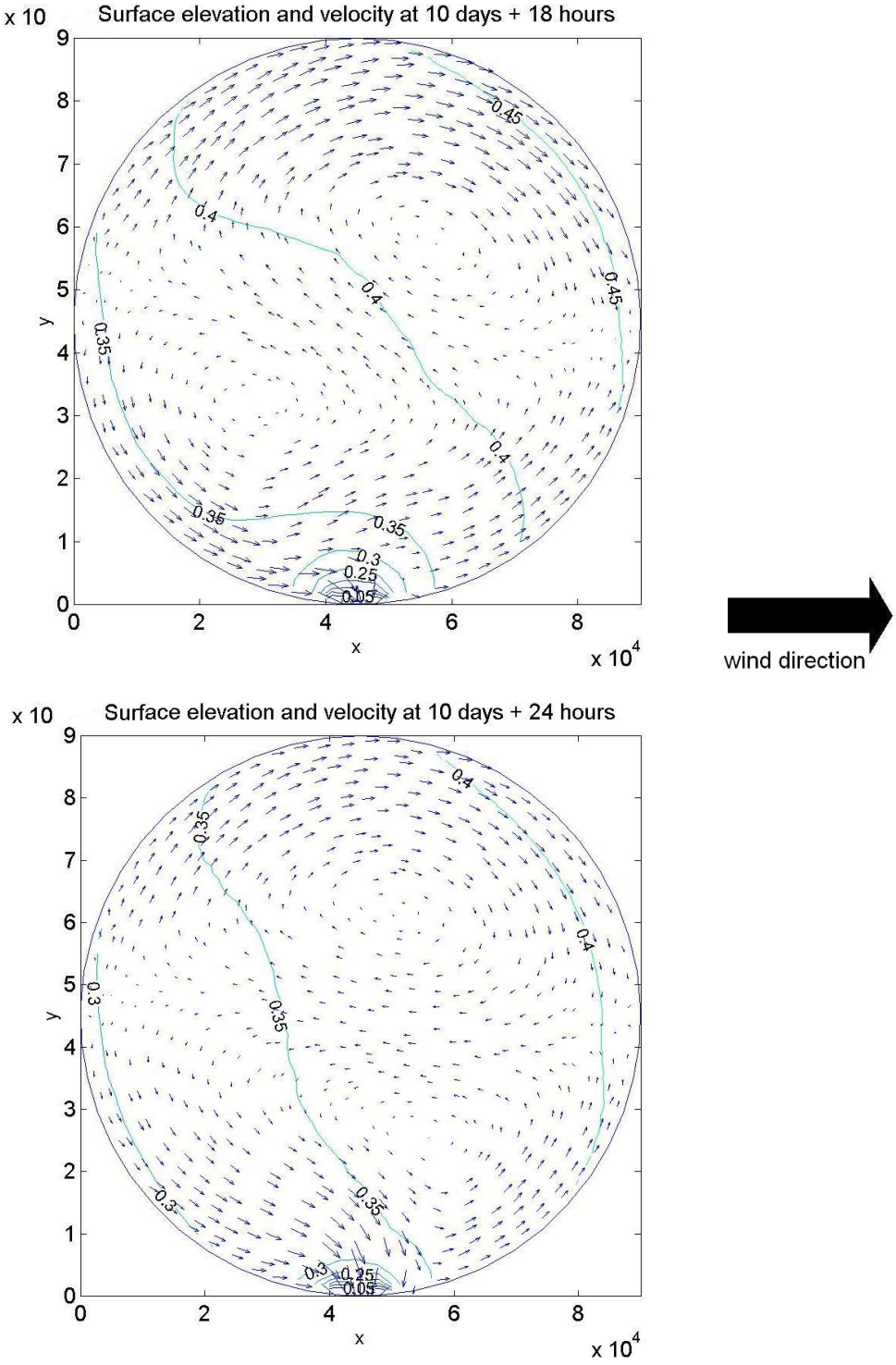


Figure 6.12: c) Surface elevation contours and velocity profile for periodic boundary conditions and a westerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (18, 24 hours).

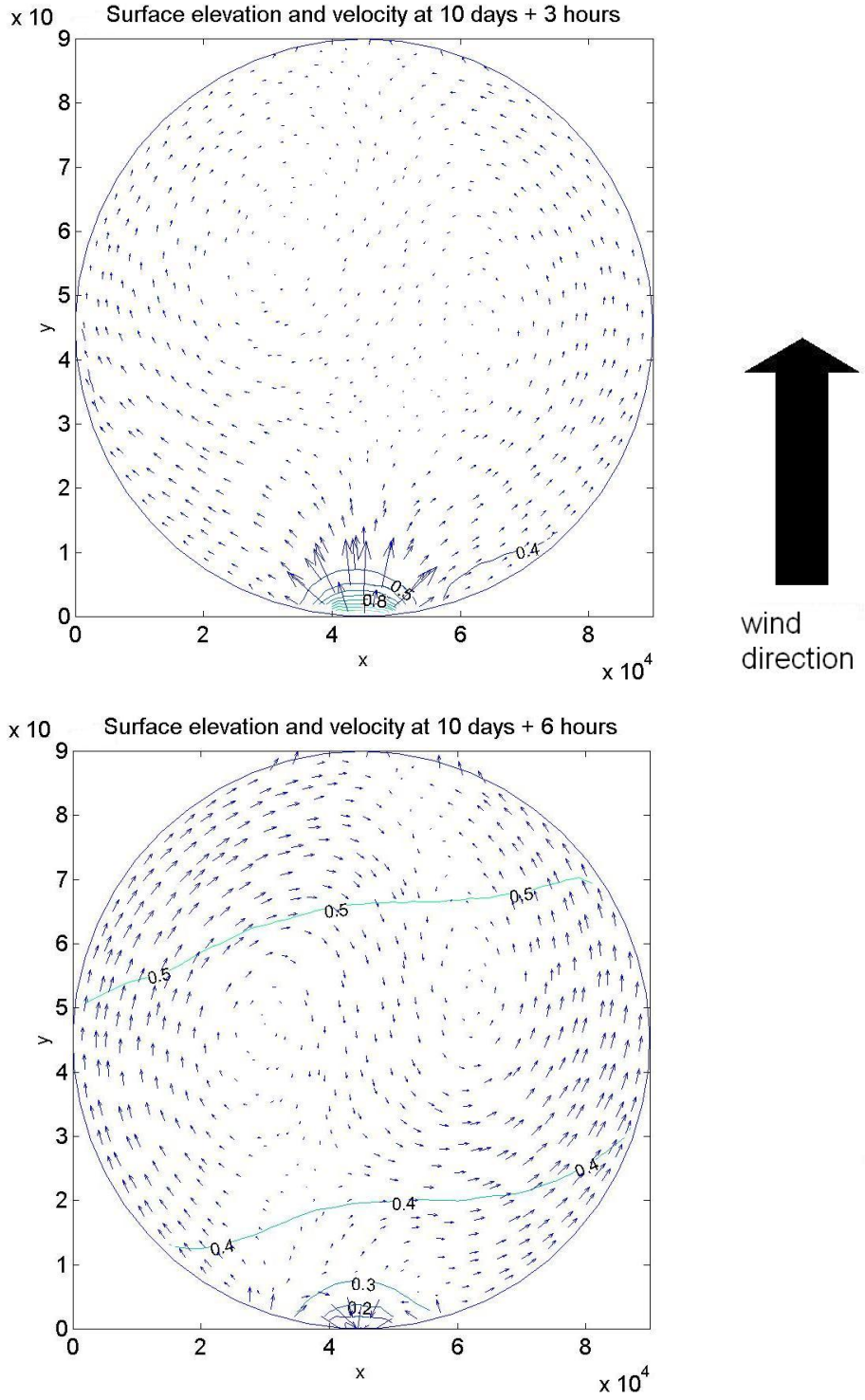


Figure 6.13: a) Surface elevation contours and velocity profile for periodic boundary conditions and a southerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (3, 6 hours).

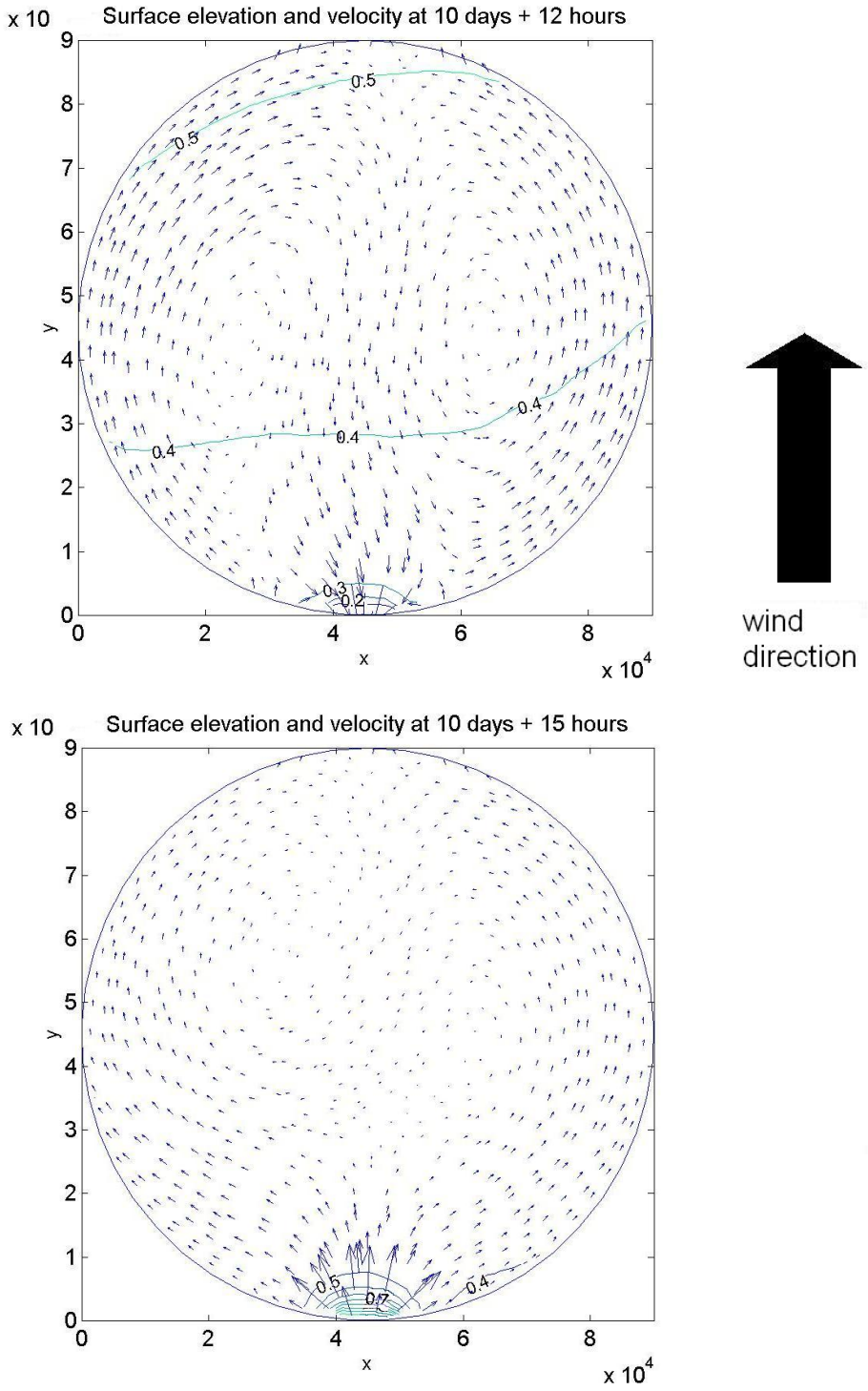


Figure 6.13: b) Surface elevation contours and velocity profile for periodic boundary conditions and a southerly wind of 5ms^{-1} on a circular domain of 2484 elements (12, 15 hours).

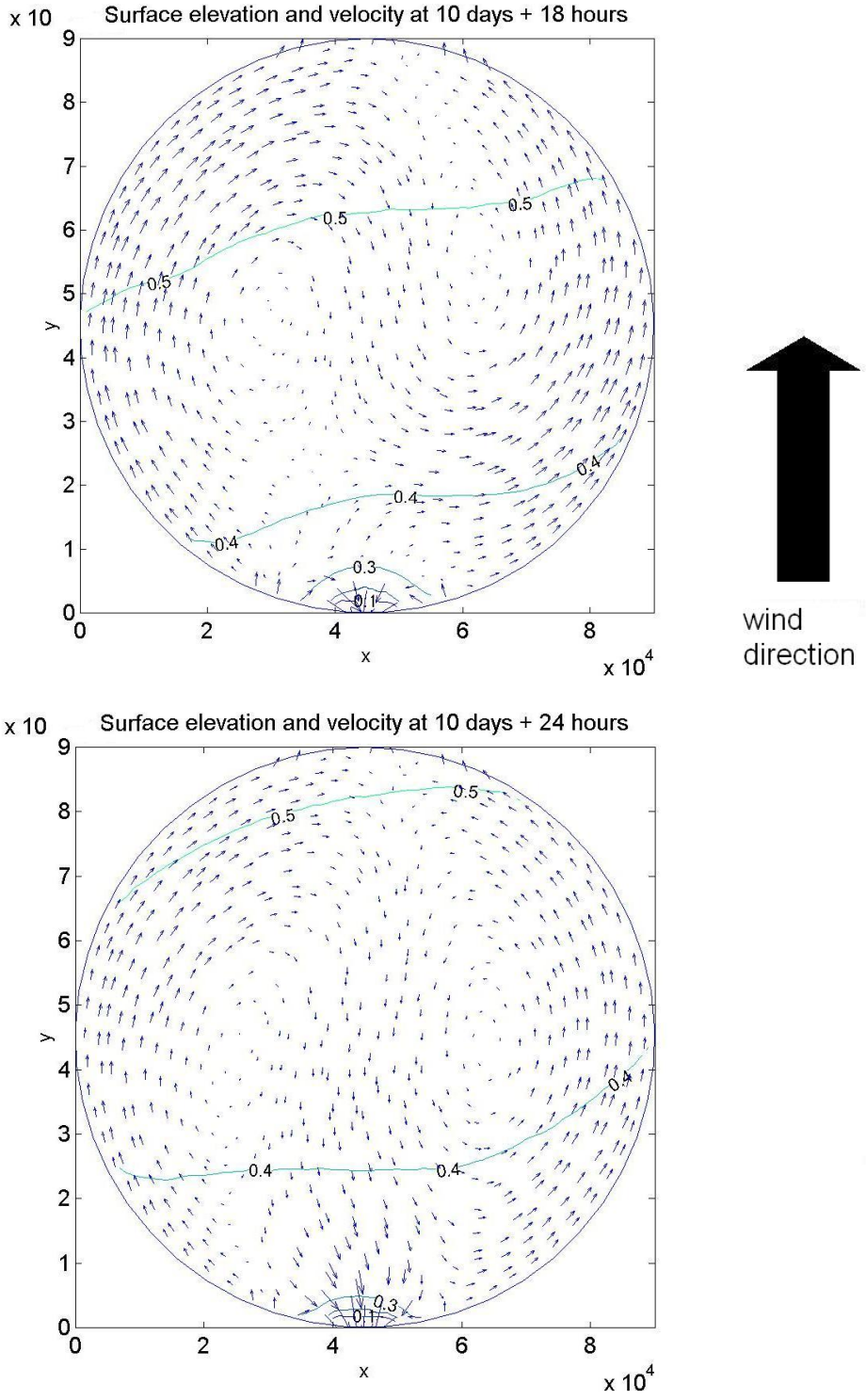


Figure 6.13: c) Surface elevation contours and velocity profile for periodic boundary conditions and a southerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (18, 24 hours).

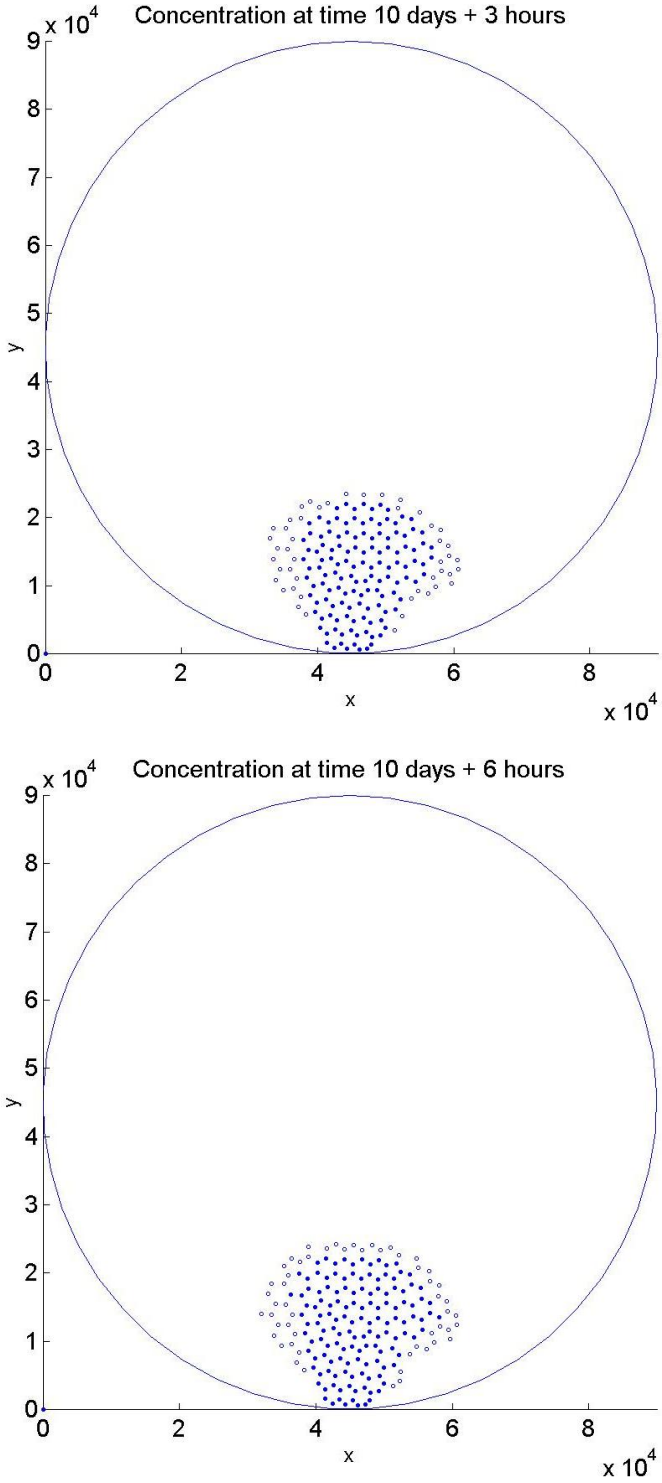


Figure 6.14: a) Scatter of concentration for periodic boundary conditions on a circular domain of 2484 elements (3, 6 hours).

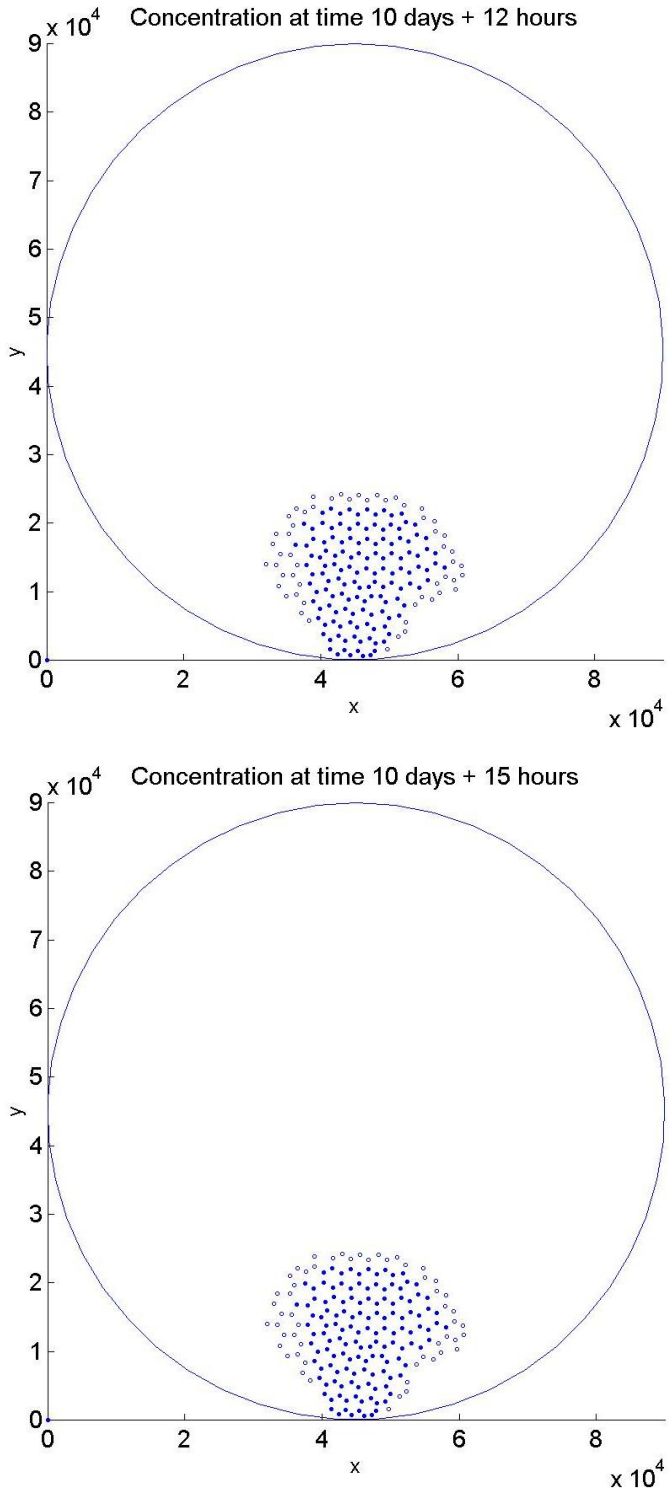


Figure 6.14: b) Scatter of concentration for periodic boundary conditions on a circular domain of 2484 elements (12, 15 hours).

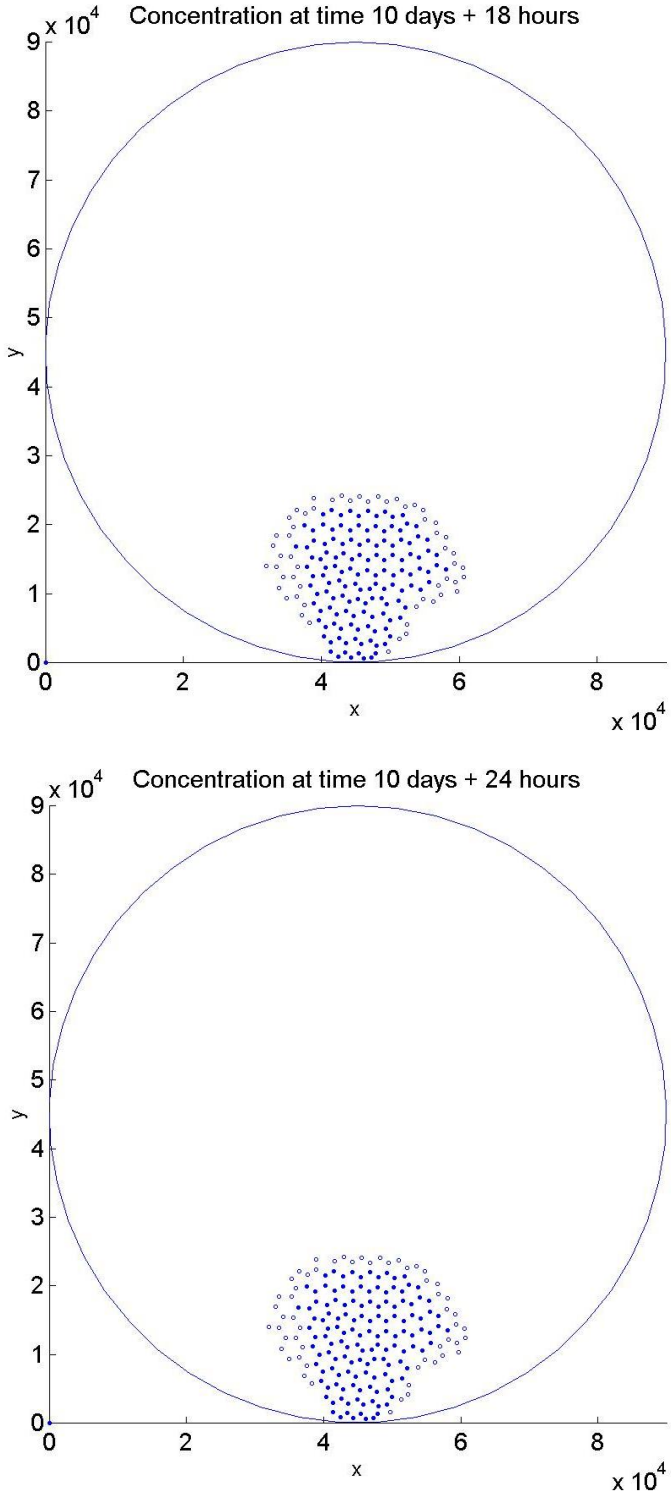


Figure 6.14: c) Scatter of concentration for periodic boundary conditions on a circular domain of 2484 elements (18, 24 hours).

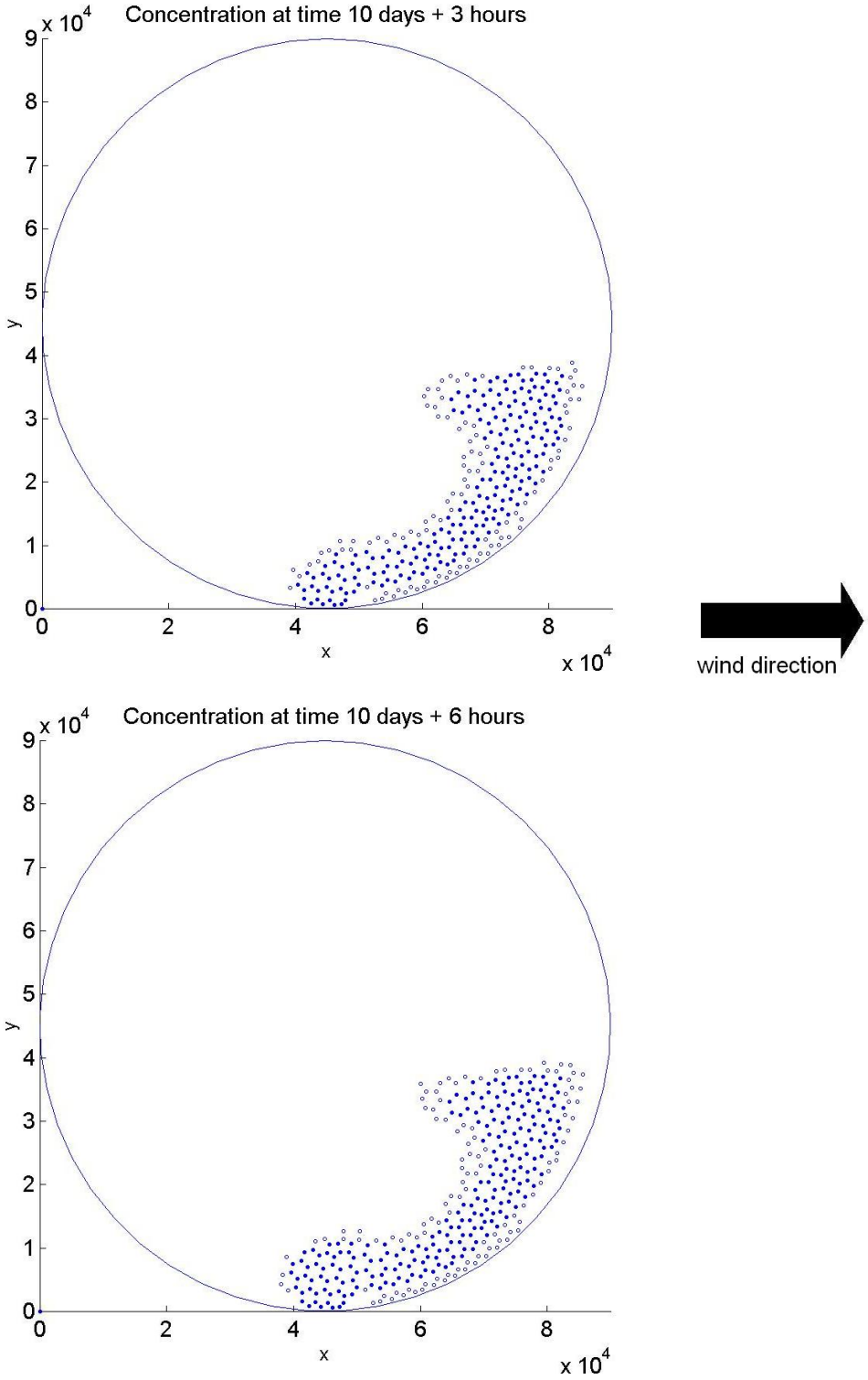


Figure 6.15: a) Scatter of concentration for periodic boundary conditions and a westerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (3, 6 hours).

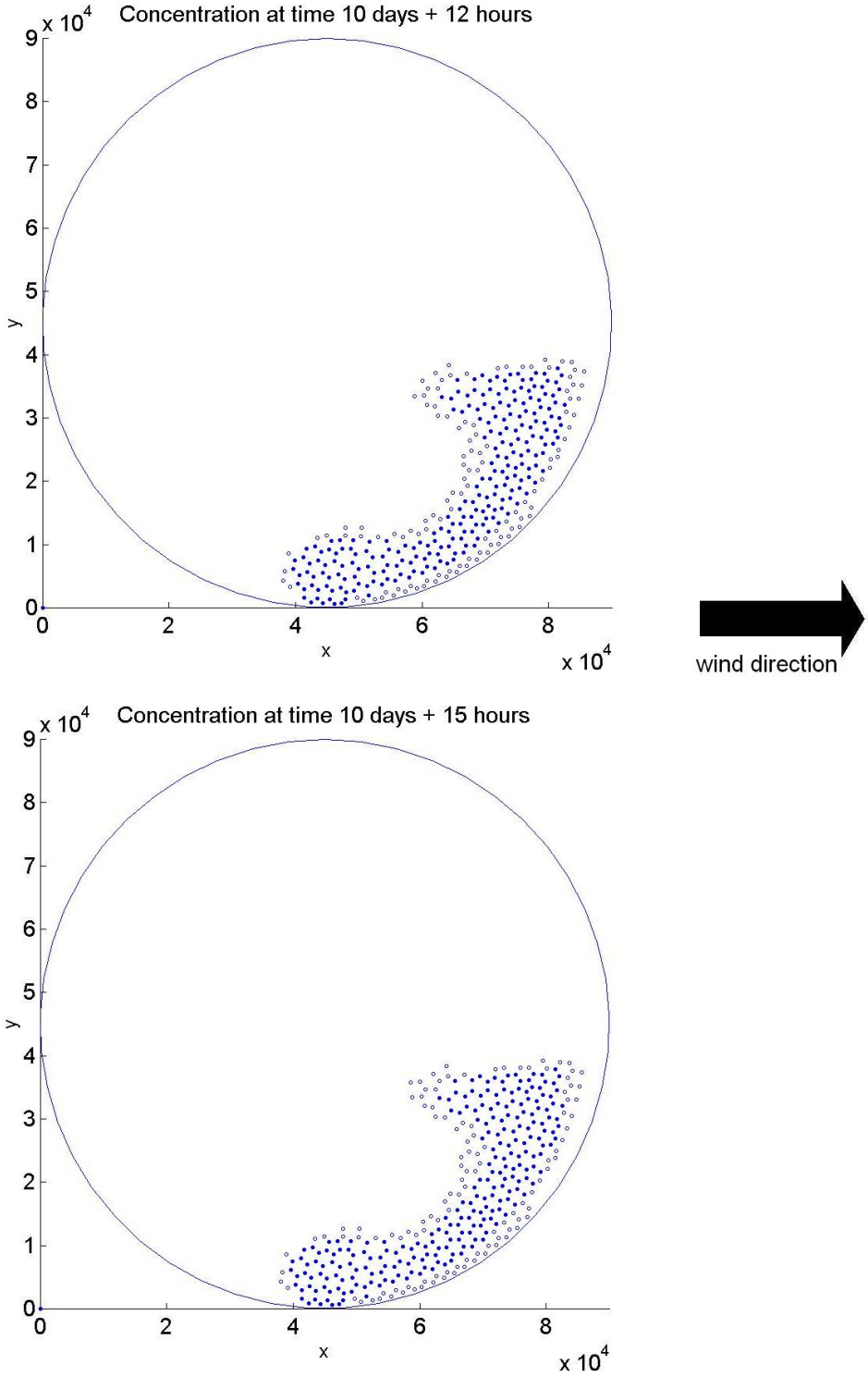


Figure 6.15: b) Scatter of concentration for periodic boundary conditions and a westerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (12, 15 hours).

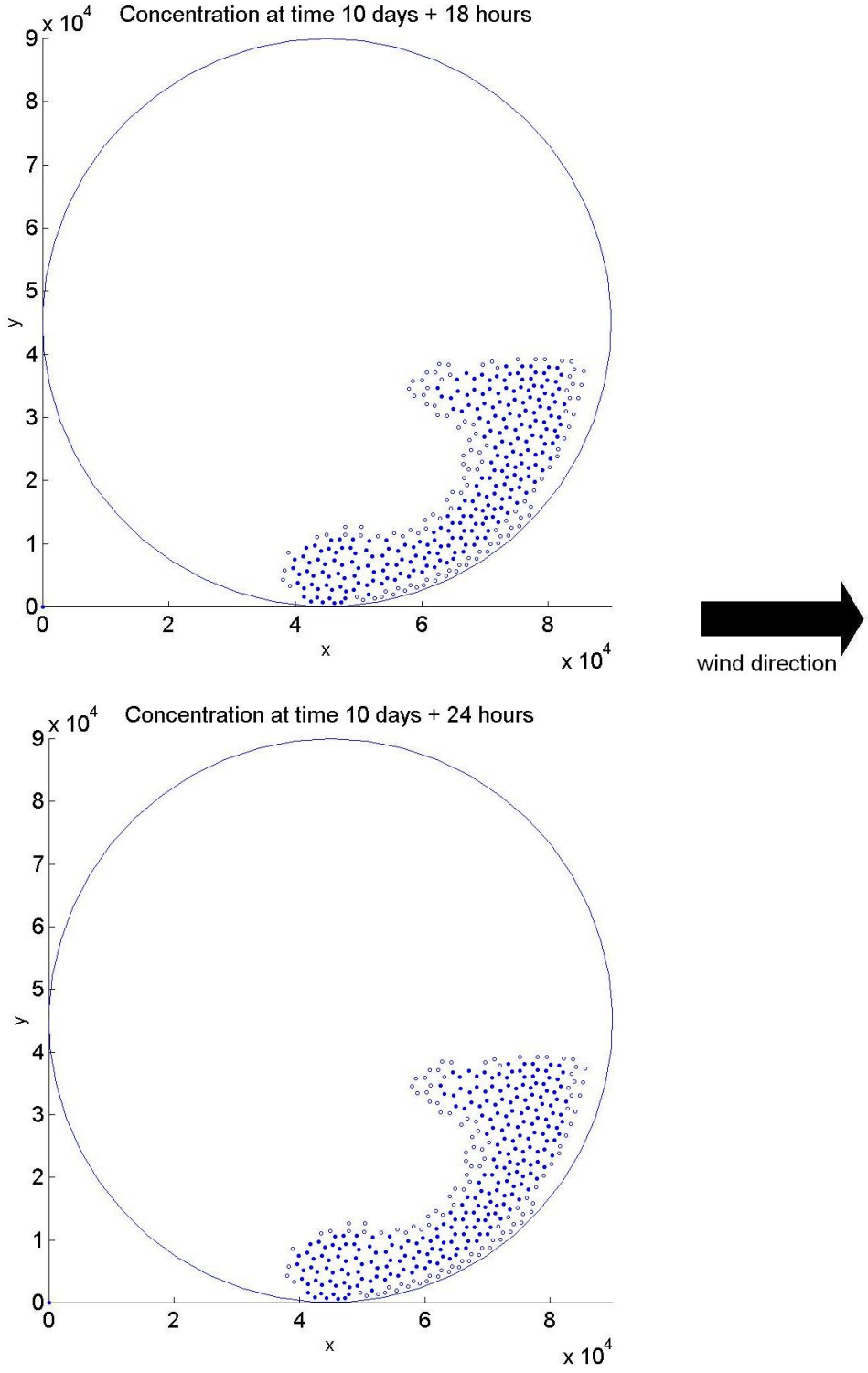


Figure 6.15: c) Scatter of concentration for periodic boundary conditions and a westerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (18, 24 hours).

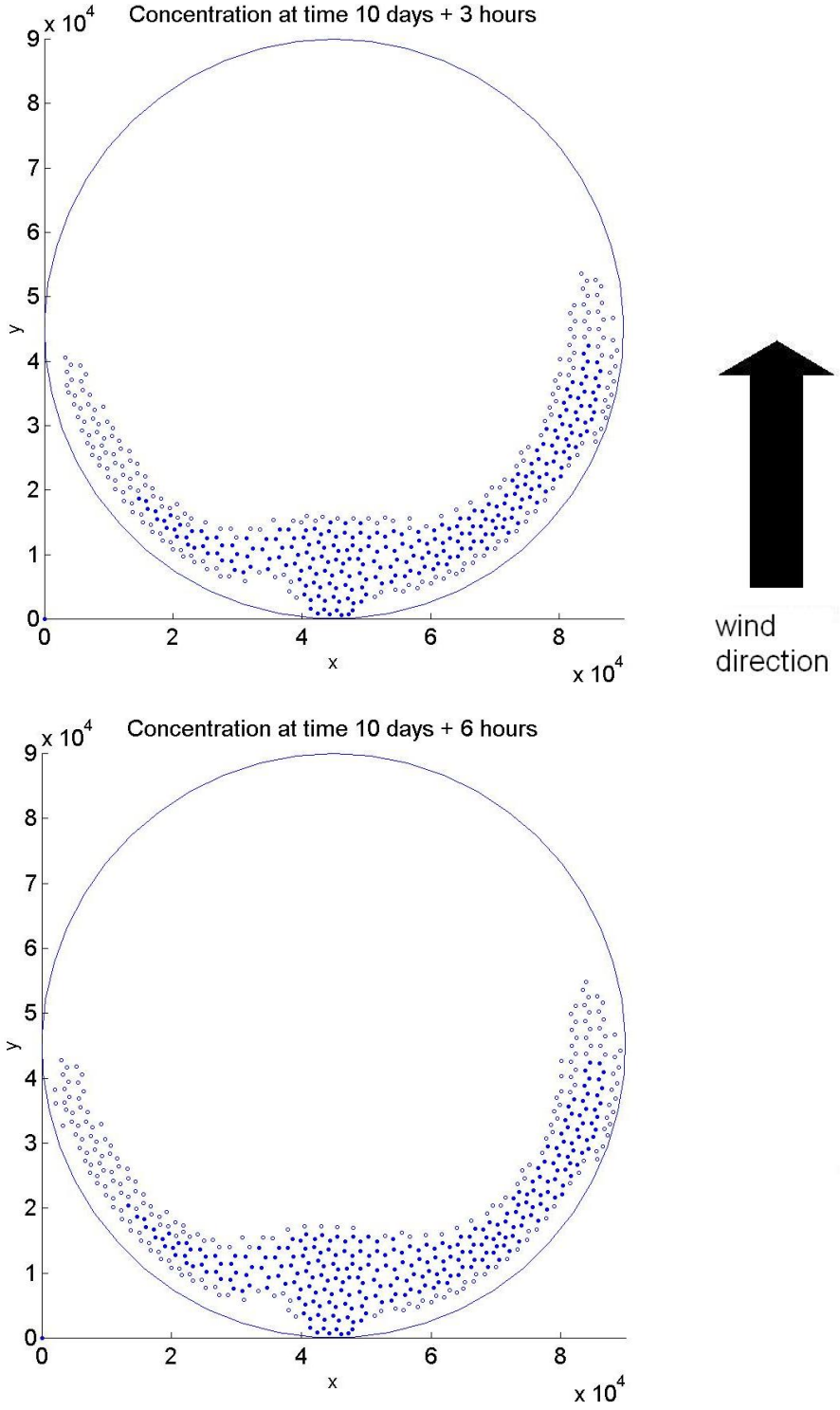


Figure 6.16: a) Scatter of concentration for periodic boundary conditions and a southerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (3, 6 hours).

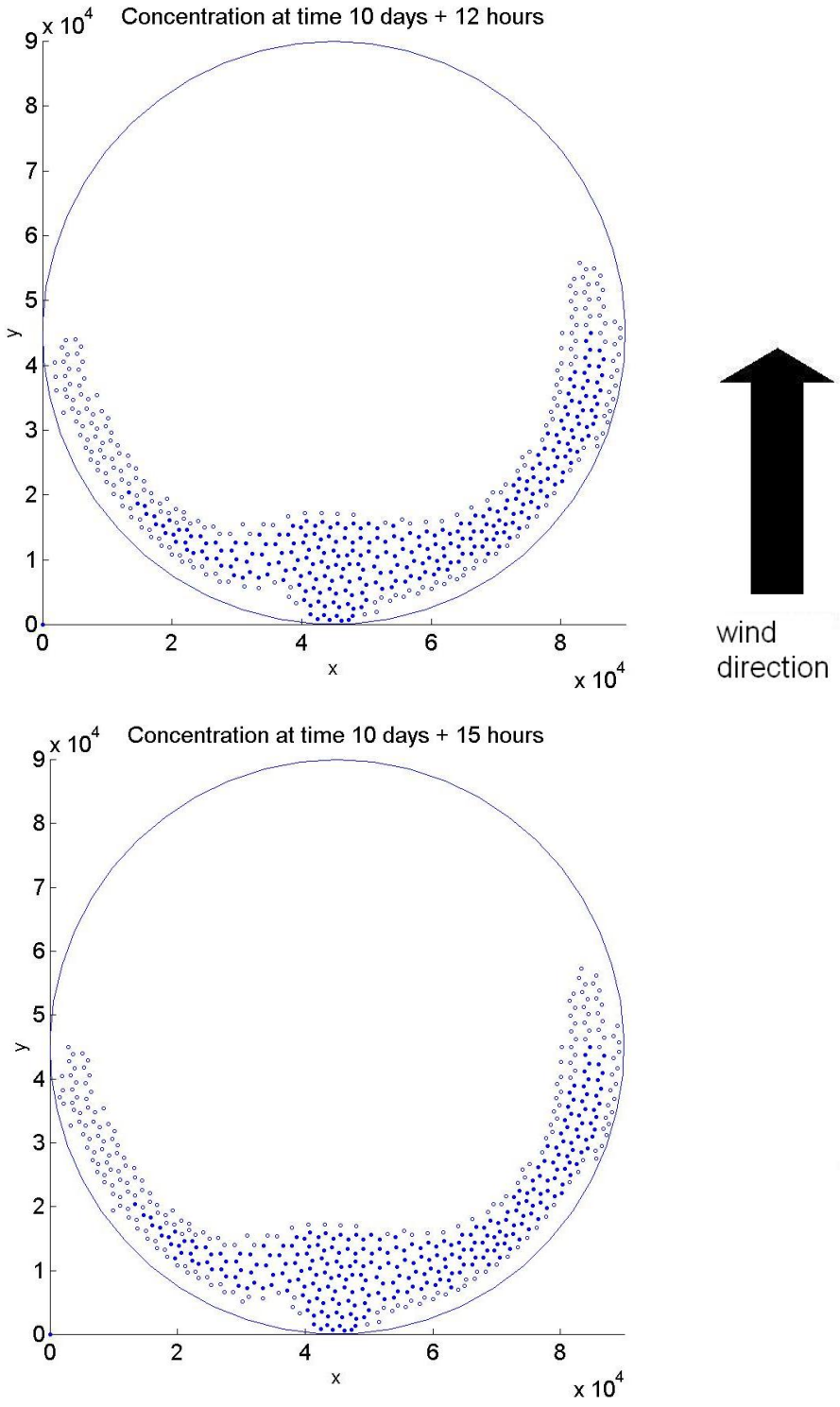


Figure 6.16: b) Scatter of concentration for periodic boundary conditions and a southerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (12, 15 hours).

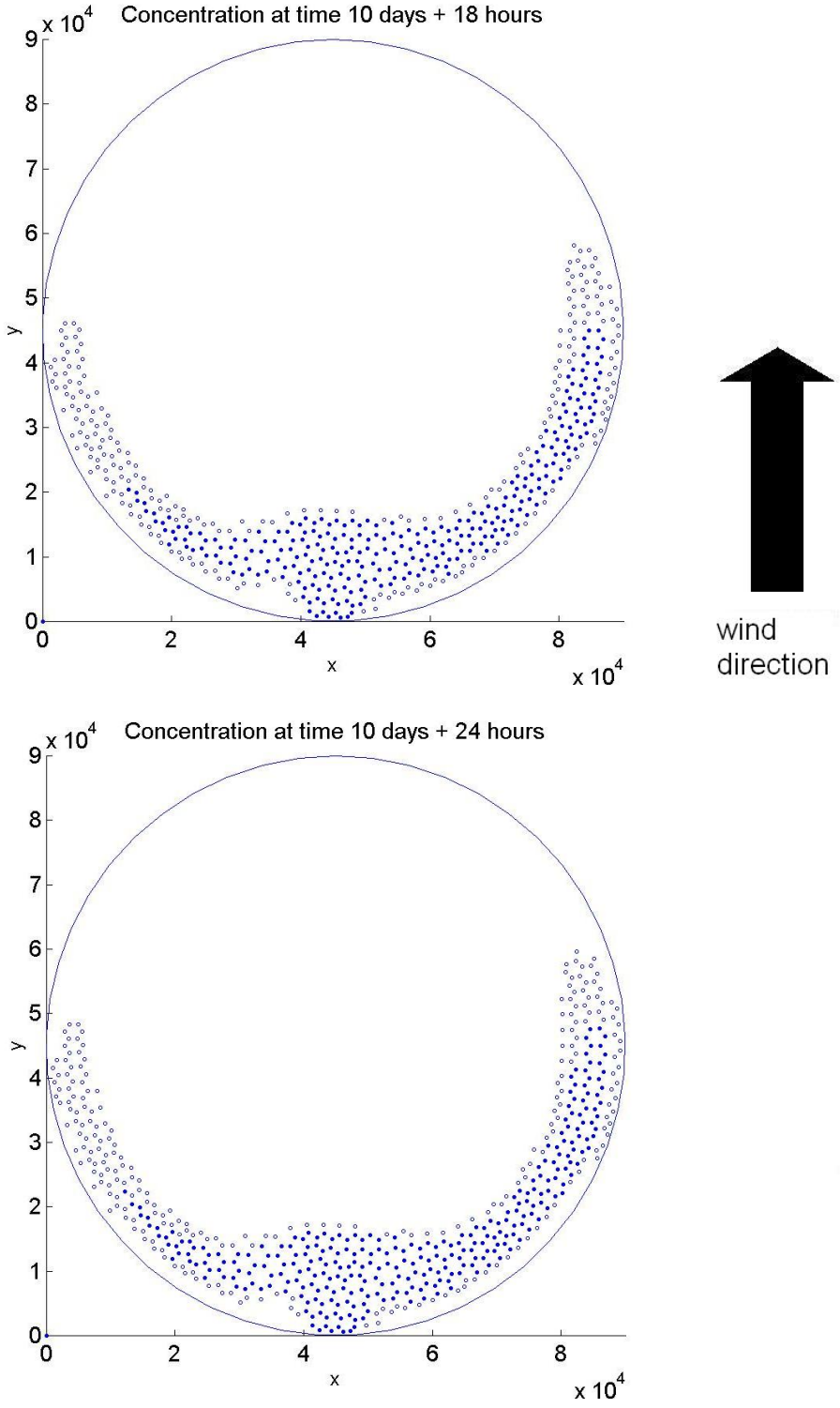


Figure 6.16: c) Scatter of concentration for periodic boundary conditions and a southerly wind of $5\,\mathrm{ms}^{-1}$ on a circular domain of 2484 elements (18, 24 hours).

6.3 Summary

The focus of this chapter has been on calibration in a qualitative sense. As we were testing generic domains, no field data were available, and to perform experiments was beyond the scope of this study. However, all the results demonstrate a good match with what we expect, and a general agreement was found with the simplified asymptotic result in Chapter 3 that the effect of the tidal boundary condition appears at $\mathcal{O}(\varepsilon)$.

The main points in this chapter are:

- *The newly-implemented tidal boundary condition was found to give the correct behaviour.*
- *Coriolis force was shown to be important in influencing the flow for a domain of this size, especially in the case of the square where bedslope effects are negligible. In the case of the circular basin, the two areas of recirculation predicted in Chapter 5 were reproduced, and showed the influence of the Coriolis force.*
- *The wind was found to affect flow velocity and surface elevation in the circular basin much more than in the square, which is explained by the result from Larsen (2007) that areas of higher bed friction limit tidal effects.*
- *The wind was shown to affect the transport of contaminant. In the square, the direction of the wind determined the extent of this influence because the tide was relatively strong due to the small bed friction. On the other hand, the wind altered the contaminant profile in the circular basin much more because the area the contaminant occupies is much more subject to shear and tidal effects are weaker.*
- *Average velocities were generally slower in the circle because of the shallow areas of higher friction where more energy is dissipated.*

We are now in a strong position to apply this model to a 'real' domain, *i.e.* using bathymetric data of Moreton Bay to simulate the flows in this bay.

CHAPTER 7

Application to Moreton Bay

Bays are subject to many stresses, including aquacultural (*e.g.* oyster farming), fisheries, recreation, sewage inputs and marine transport (Greenberg *et al.* (2005)). In particular Moreton Bay is a popular tourist destination, and is home to various wildlife such as sea turtles and dugong. On the other hand, it is a source of sand for building materials so it is periodically dredged, and as the city of Brisbane continues to expand, leading to greater industrial and agricultural activity, there is an increased likelihood of pollutants entering the Bay.

In this chapter *Riemann2D* is used to simulate flows in Moreton Bay, using a tidal forcing condition at the ocean boundary. Results are calibrated with field measurements. The advection of a contaminant released at the Brisbane river mouth is studied according to the tidal forcing and different wind speeds to evaluate the effects of these forces on transporting the contaminant.

7.1 Moreton Bay

Moreton Bay is located to the east of Brisbane in Southeast Queensland. It is around 90 km from north to south, and has a maximum width of 30 km. There are two large sandy islands to the east of the Bay, Moreton Island and North Stradbroke Island, as well as South Stradbroke Island to the southeast, which means that the Bay is partially enclosed. To the north of the Bay there is an outlet to the ocean of about 14.5 km across and there is another small outlet (about 1.6 km, You (2005b)) between Moreton and North Stradbroke islands, known as southern

passage. Most of the water in the Bay is between 10 and 20 metres deep. Moreton Bay is fed by five rivers, the largest of which is Brisbane river. The other four are Caboolture river in the north, Pine river to the north of Brisbane, and the Logan and Coomera rivers running into the south of the Bay (see map in Figure 7.1).

The Brisbane river is a major source of polluted water. As well as being the largest of Moreton Bay's rivers, the Brisbane river also has a large, well-populated catchment area, which means it has greater potential than others for delivering unwanted pollutants (Olley *et al.* (2006)). Since the area of Moreton Bay is of great ecological significance (as explained by Dunn *et al.* (2007), Haynes (2001), Hodge *et al.* (2005)), it is important to be able to predict the fate of a pollutant, in order for well-informed management decisions to be made for the protection of the environment.

7.2 Domain of calculation

A set of data on the bed elevation of Moreton Bay was provided by Chris Matthews at Griffith University, Brisbane. The area is shown in a satellite image in Figure 7.1, and Figure 7.2 shows the data plotted using Argus, with colour corresponding to elevation, and the mesh superimposed. The islands in the southern part of the Bay are excluded from the domain of calculation and are shown as hashed areas in Figure 7.2.

7.2.1 Mesh generation

Figure 7.2 shows the mesh of 2359 elements on which all the tests in this chapter have been made. Argus ONE® meshmaker is used to import the data describing the bathymetry of Moreton Bay. The meshfile is made by drawing around the edge of the land and around the islands, then automatically generating a triangular mesh that fits around the irregular edge of the Bay. The total area of the mesh according to Argus is approximately $1\,180\,\mathrm{km}^2$, which agrees with You (2005b) ($1\,130\,\mathrm{km}^2$) and the smallest element has an area of $0.04\,\mathrm{km}^2$, the largest $0.8\,\mathrm{km}^2$ and an average size of about $0.5\,\mathrm{km}^2$.

Figure 7.1: Plan view of Moreton Bay with some locations marked. Courtesy of Google maps.

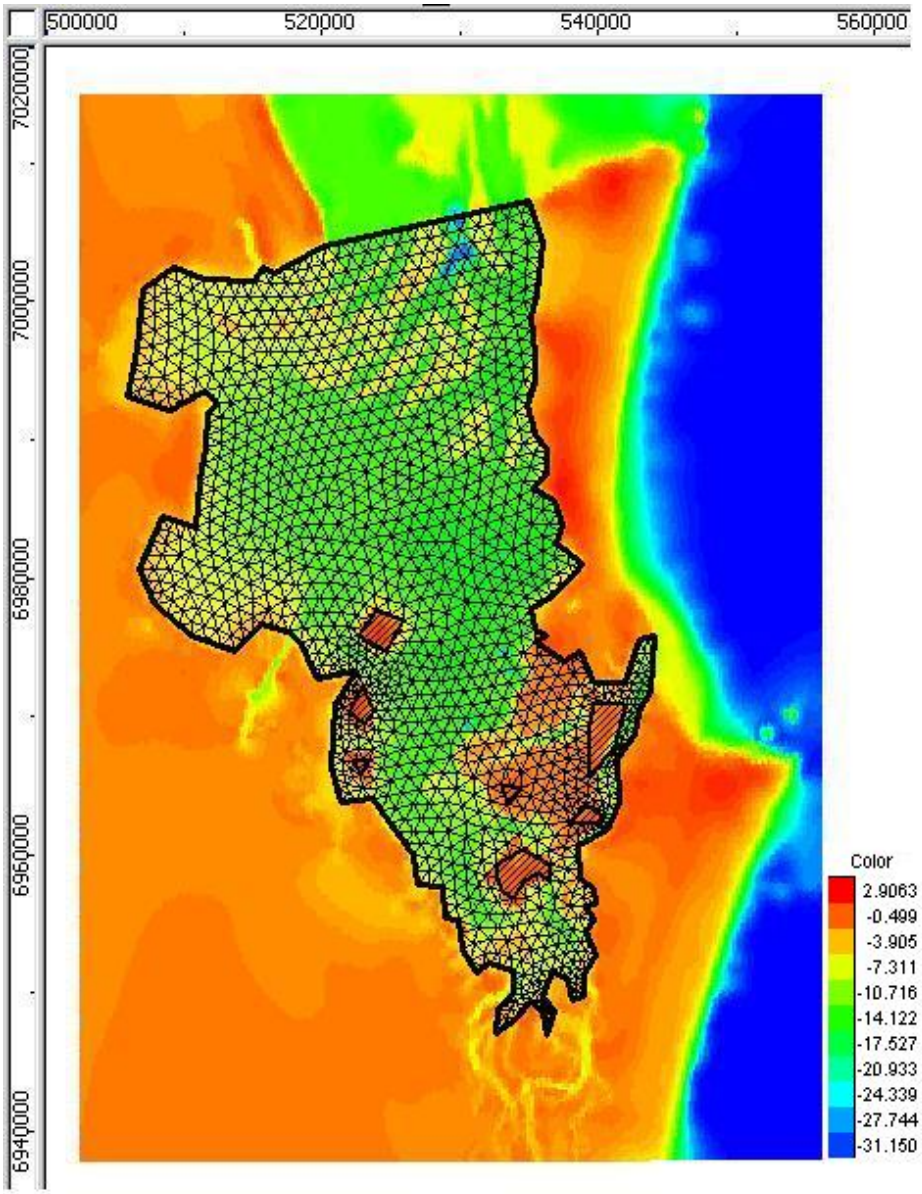


Figure 7.2: Mesh used for calculations of the following tests. Orange/red colours represent land above sea level, blue/green/yellow areas show the land with negative elevation, *i.e.* below sea level. Islands are shown with solid boundaries and hashed shading. Mesh is composed of 2359 triangular elements, and was generated using Argus meshmaker.

Point in harbour	AHD	Mean sea level	(Mean sea level)-(AHD)
Pinkenba (mouth of Brisbane River)	1.24	1.27	+0.03
Nudgee Beach and Crib Island	1.31	1.19	-0.22
Woody Point	1.23	1.15	-0.08
Beachmere	1.26	1.21	-0.05
Bongaree	1.10	1.05	-0.05
Toorbul	1.10	1.13	+0.03
Dunwich	1.30	1.22	-0.08
Russell Island	1.39	1.33	-0.06
Redland Bay	1.35	1.41	+0.06
Manly	1.29	1.27	-0.02

Table 7.1: Table showing data of Australian height datum (AHD) and mean sea level (metres) for a number of places around Moreton Bay. Reproduced from the Official Tide Tables of Australia.

7.2.2 Australian Height Datum

The data that were sent from Griffith University take the plane $z = 0$ at what is known as the Australian Height Datum (AHD). The bed elevation therefore has negative values, and anything above AHD (positive values) is taken to be land, being either the islands or the mainland. The AHD is based on mean sea levels taken in 1966-68 at thirty points around Australia. In 1971 it was adopted by the National Mapping Council as "the datum to which all vertical control for mapping is to be referred" [1]. Today the AHD is generally lower than the mean sea level due to rising sea levels, although may be locally greater because of sedimentation, human development or continental drift (ICSM (2006)). A copy of the Official Tide Tables and Boating Safety Guide 2007 [2] for harbours around Australia gives the AHD and the present mean sea level for a number of points around the Bay, a sample of which is reproduced in Table 7.1. The corresponding locations are shown in Figure 7.1.

It can be seen from Table 7.1 that the average AHD is 1.26 m and the average mean sea

1. www.ga.gov.au/geodesy/datums/ahd.jsp
2. Copyright Commonwealth of Australia 2005

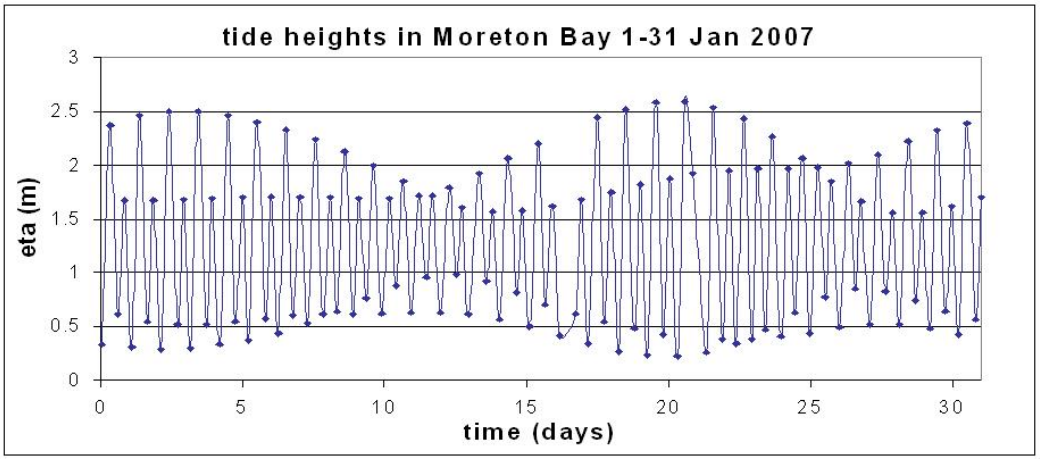


Figure 7.3: Tide heights of Moreton Bay for January 2007, measured at Brisbane river mouth. Produced using data from National Tidal Centre, Bureau of Meteorology, Commonwealth of Australia.

level is 1.22 m. Despite the inhomogeneities of the surface level in this model we assume the level is constant, which is a valid approximation for the level of accuracy required.

7.2.3 Tides of Moreton Bay

Figure 7.3 shows the surface elevation in Moreton Bay, measured at Brisbane river mouth, during the period 1-31 January 2007. The time between high tide and low tide is approximately six hours so the tide is semidiurnal. It can be seen in this graph that on 4 January and on 20 January there was a new moon; these points correspond to periods of the highest high tides and the lowest low tides (a spring tide, just after full and new moons). Around 11 and 26 January, the moon was midway through its cycle, and these periods have the lowest high tides and the highest low tides (neap tides).

The average low tide from the data in Figure 7.3 is 0.54 m, and the average high tide is 1.99 m. Thus the average range is 1.45 m, which varies between 0.8 m and 2.3 m.

7.3 Calibration of *Riemann2D*

The tidal current and tide level (surface elevation) have been measured by You (2005b) in the most northwest embayment of Moreton Bay where Caboolture river discharges, marked on the map as Deception Bay. These measurements are used to calibrate results from *Riemann2D* .

7.3.1 Calibration of tidal currents

The field measurements made by You (2005b) are reproduced in Figure 7.4. A curve of best fit was made to fit with the data (11-19 March). The best fit curves are made by using the functions

$$\eta = -0.1 + 0.7\sin(\pi t/6.6 + 7.5), \tag{7.1}$$
$$|u| = \sqrt{u^2 + v^2} = 0.56 + 0.26\sin(\pi(t - 4.5)/2.95) + 0.15\cos(\pi t/5.5),$$

for the surface elevation and the tidal current, respectively (time in units of hours). Note that these measurements are likely to have been affected by the wind, although wind speeds were not noted, and we have no way of knowing how much the wind affected the flow velocity. So they must be considered as specific, not general data. Nevertheless, they provide a reasonable basis for modelling. You (2005b) mentions that there was a storm on 26 March, which produced large current speeds, so we avoided using measurements from near this date.

The calibration tests were done using the Moreton Bay mesh of 2359 elements, with minDepth 1×10^{-1} m, Coriolis -6.6×10^{-5}, Manning's $n = 0.03\,\mathrm{m}^{-1/3}\mathrm{s}$, and a fixed timestep of dt $= 0.25$ s. Tide-like boundary conditions were set at the ocean boundary so that the surface elevation is set in the ghost cell as in (7.1) and the flux was prescribed by setting

$$hu_{ghost} = (0.56 + 0.26\sin(\pi(t - 4.5)/2.95) + 0.15\cos(\pi t/5.5))h/\sqrt{2}, \tag{7.2a}$$

$$hv_{ghost} = (0.56 + 0.26\sin(\pi(t - 4.5)/2.95) + 0.15\cos(\pi t/5.5))h/\sqrt{2}, \tag{7.2b}$$

For comparison, a second test was made where the flux boundary conditions were reflective, as was done in Chapter 6. In both cases, the Courant number was around 0.013, and the simulation was run for 9 days, which took around 31 hours and 30 minutes.

The results near initial time are influenced by the fact that flow was initially still. The simulation was run for eight days before any data were recorded to remove any phenomena that may be related to the water's initial state.

The surface elevation and tidal current from these simulations are plotted in Figures 7.5 (prescribed flux) and 7.6 (reflective flux), and show good agreement with the measurements of You (2005b). The good agreement with reflective boundary conditions suggests the results from the previous chapter are realistic, even though we were not able to calibrate results for those cases.

Both prescribed and reflective flux boundary conditions demonstrate good agreement with surface elevation. The behaviour of the tidal current is more difficult to capture than the surface elevation. In the literature, Castro *et al.* (2004) has studied the 1D SWE for two fluid layers in a channel applied to the Strait of Gibraltar. Tidal forcing is applied to the steady-state solution that is obtained. The model is forced at the open boundaries with the four main tidal components using a cosine function that is interpolated to give a best fit. The maximum differences between simulated and observed elevation do not exceed 3 cm (5%) for the M_2 and S_2 components and the phase for M_2 was correct to two decimal places. Abdennadher and Boukthir (2006) use the ROMS (Regional Ocean Modelling System), which is a free-surface oceanic model that includes temperature and salinity to model the area between the Tunisian Shelf and the Strait of Sicily. The four open boundaries are forced with a cosine function that include the five major tidal components. The results obtained show that the M_2 component is within 2 cm for the amplitude, or 5%, and 14° for the phase of the observed measurements. This compares favourably with other results for the same area, and the otherwise slightly poor agreement is thought to be due to uncertainties in the bathymetric data. Liang *et al.* (2006) attribute their poor agreement between the velocity in the numerical results to the areas of irregular topography.

On the whole, the results with reflective flux on the boundary are slightly better, so these will be used in future tests. The magnitudes are slightly lower because no wind was modelled, as noted above.

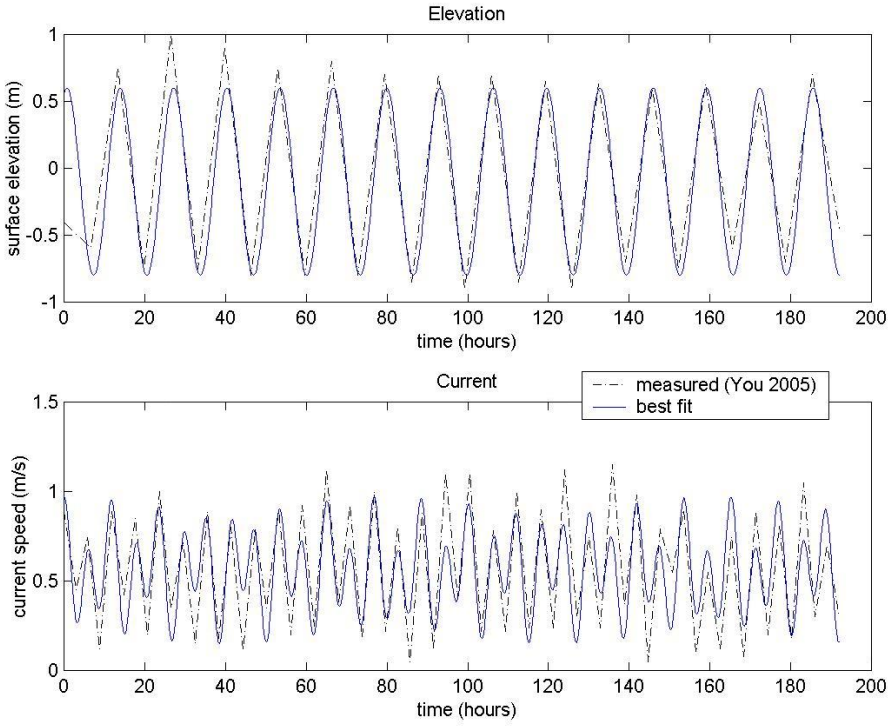


Figure 7.4: Tide current and level measured by You (2005b) between 11th and 19th March (192 hours) and best fit curves to these measurements that will be used as boundary conditions. Note that current speed is given by $|u| = \sqrt{u^2 + v^2}$.

7.3.2 Tidal and river effects on surface elevation

The test was repeated with a discharge from the Brisbane river. This was simulated by setting the element at the mouth of the river to have elevation 0.1 m at every time step, which corresponds to outflow during the dry season, according to Eyre *et al.* (1998) and Hossain *et al.* (2004). Figures 7.7a)-f) show the surface elevation and velocity profile from this test, with Coriolis, velocity vectors scaled by 0.05. The times shown correspond to high tide, mid-ebb and mid-flood tide times according to Figure 7.5.

The Coriolis effect can be seen in the northern part of the Bay, where flow is being pulled in a clockwise direction. Flow in the southern part of the Bay is interrupted by the islands there, where water is subject to high shear and quite high velocities can be generated. The argument

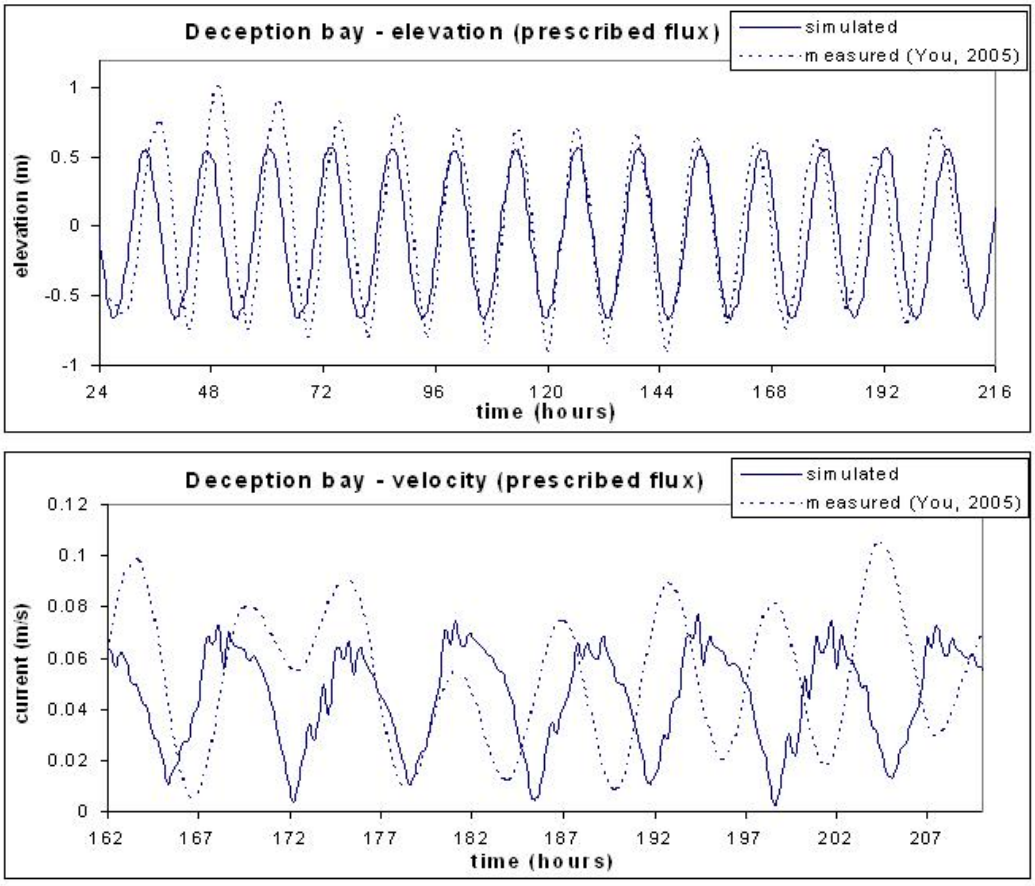


Figure 7.5: Using tide current and level measured by You (2005b) as boundary conditions in *Riemann2D* on a mesh of 2359 elements of Moreton Bay. Data plotted are those of the study site in Deception bay.

for the omission of a tidal boundary condition at the southern passage is strengthened by the results of Larsen (2007), who notes that tidal effects are weakened in areas of high bed friction.

7.4 Effect of tide on pollutants

A contaminant was released at seven and a half days at the mouth of the Brisbane river, with initial (nominal) value 0.3 kg m^{-3}. The value updated at every timestep so that the boundary condition at this cell was $hc_{ghost} = 0.3\,\text{kg m}^{-3} \times 9.14\,\text{m}$. It was found that releasing the contaminant at 7 and a half days was necessary to ensure all transients resulting from the

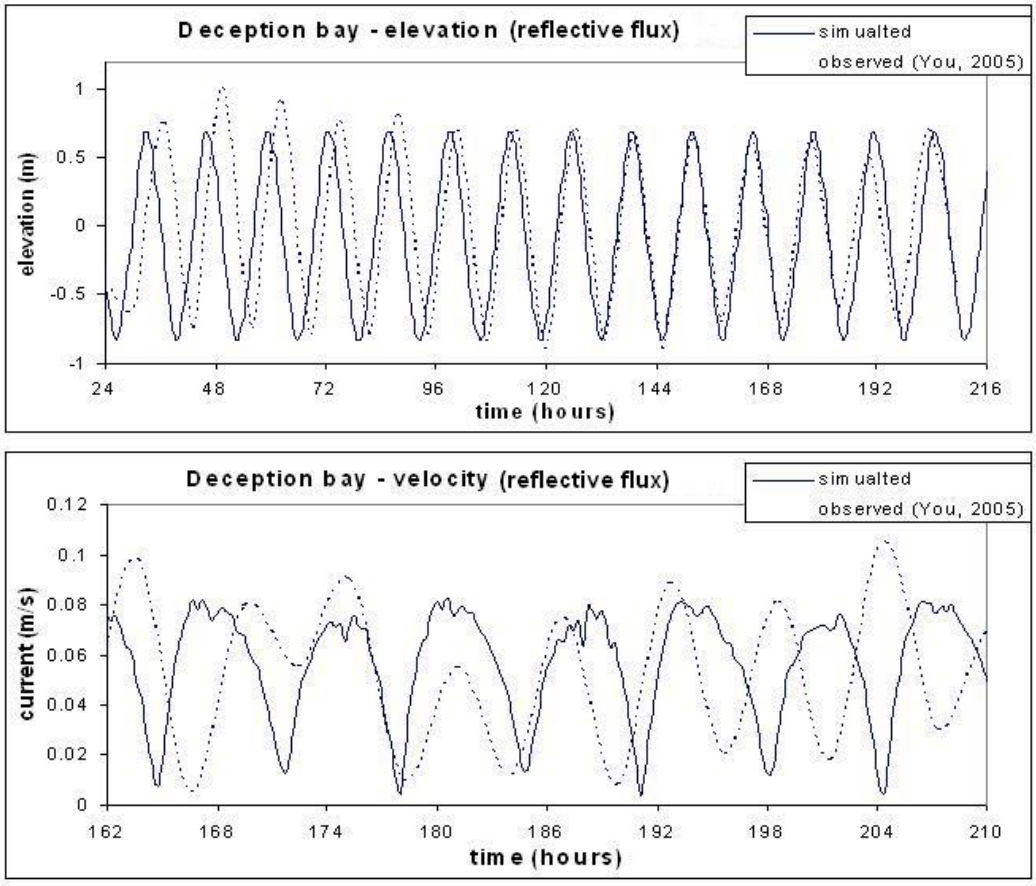


Figure 7.6: Using measurements of You (2005b) as boundary conditions for surface elevation and a reflective flux condition in *Riemann2D* on a mesh of 2359 elements of Moreton Bay. Data plotted are those of the study site in Deception bay.

initial conditions were no longer affecting the solution.

7.4.1 Contaminant release at Brisbane river

Figures 7.8a)-c) show the path of the contaminant during the 24 hours after 8 days. They show that the contaminant spreads northwest from the source as it discharges from the river mouth, into the area known as Bramble bay, because of the direction of velocity vectors (see Figures 7.7) due to the local geometry and the pull of the Coriolis force in the clockwise direction.

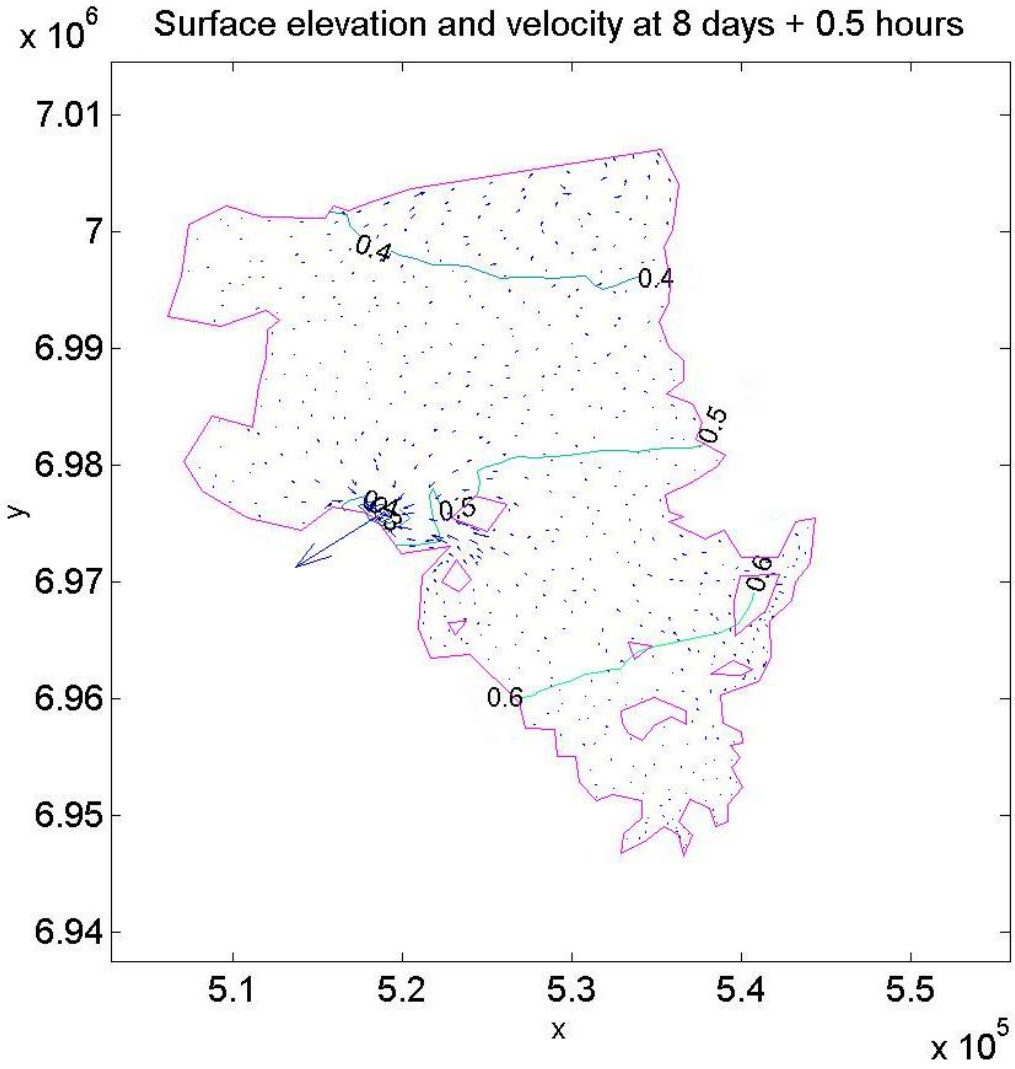


Figure 7.7: a) Surface elevation and velocity plots for periodic boundary conditions on a domain of Moreton Bay (8 days and 0.5 hours, high tide).

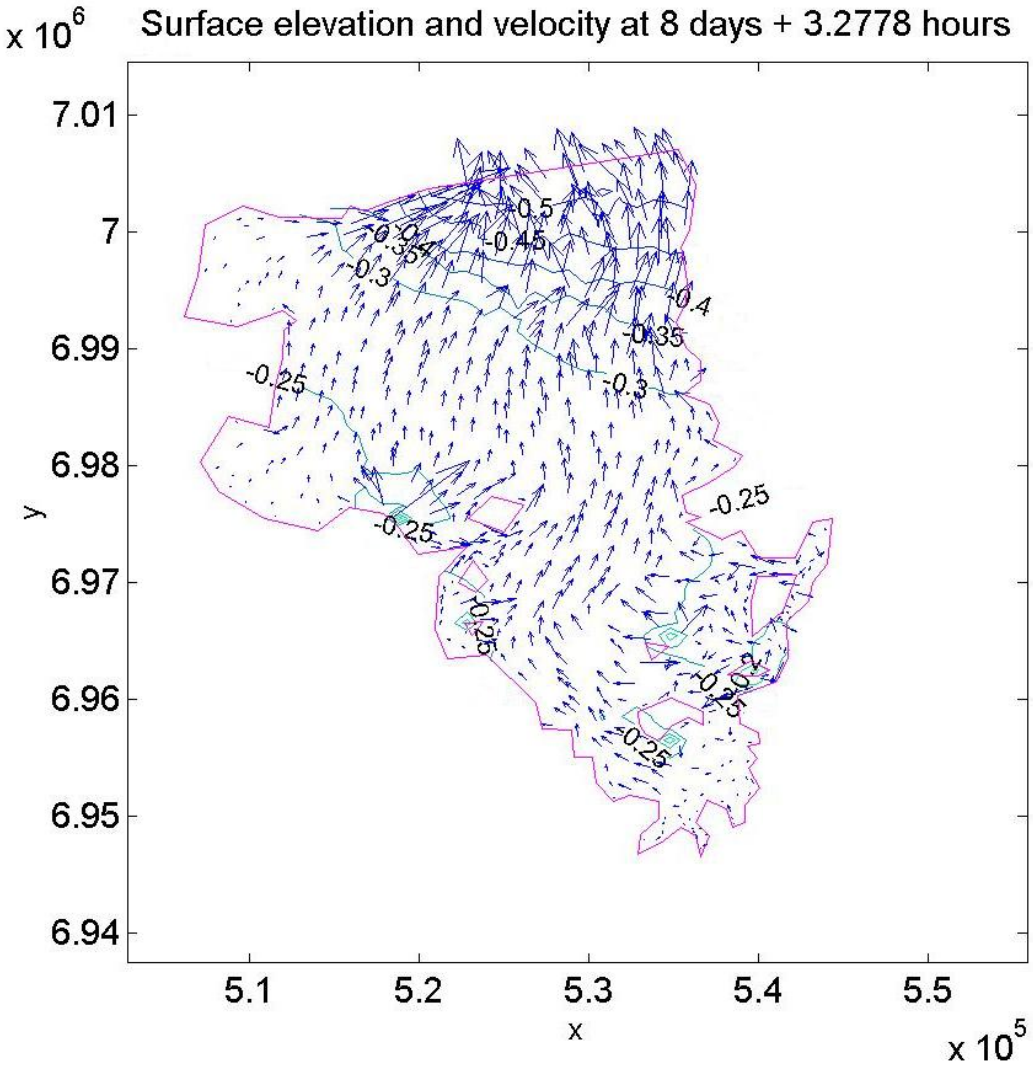


Figure 7.7: b) Surface elevation and velocity plots for periodic boundary conditions on a domain of Moreton Bay (8 days and 3 hours, mid-ebb).

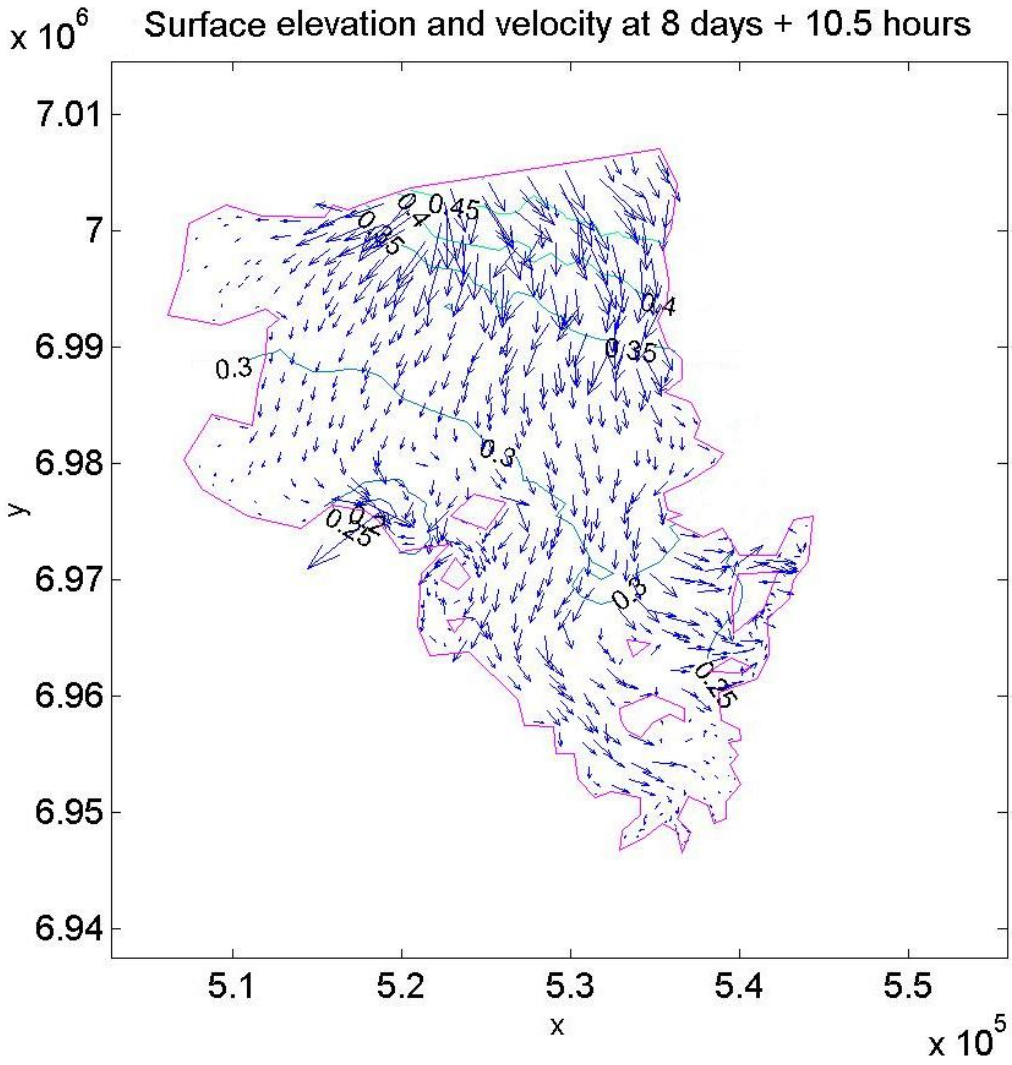


Figure 7.7: c) Surface elevation and velocity plots for periodic boundary conditions on a domain of Moreton Bay (8 days and 10.5 hours, mid-flood).

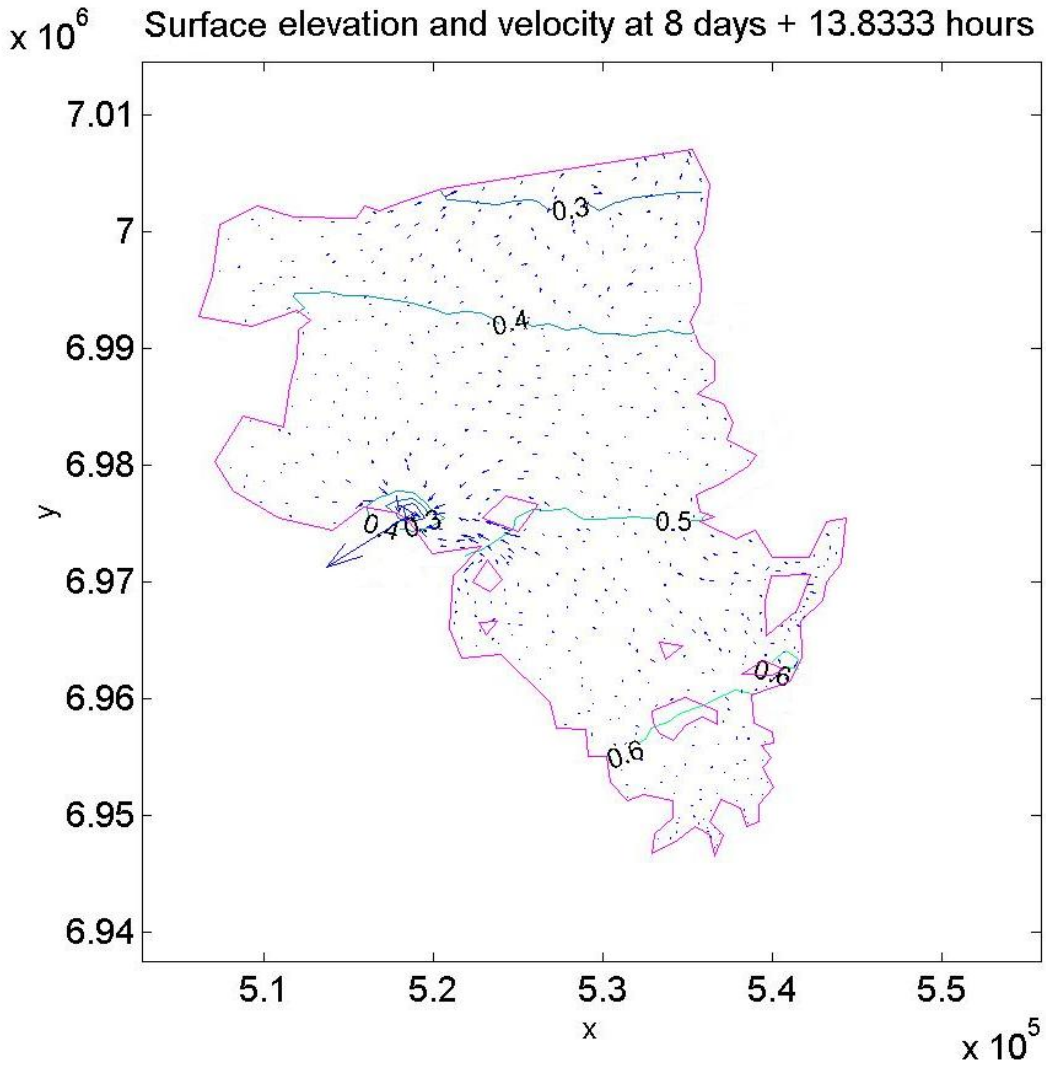


Figure 7.7: d) Surface elevation and velocity plots for periodic boundary conditions on a domain of Moreton Bay (8 days and 13 hours 50 minutes, high tide).

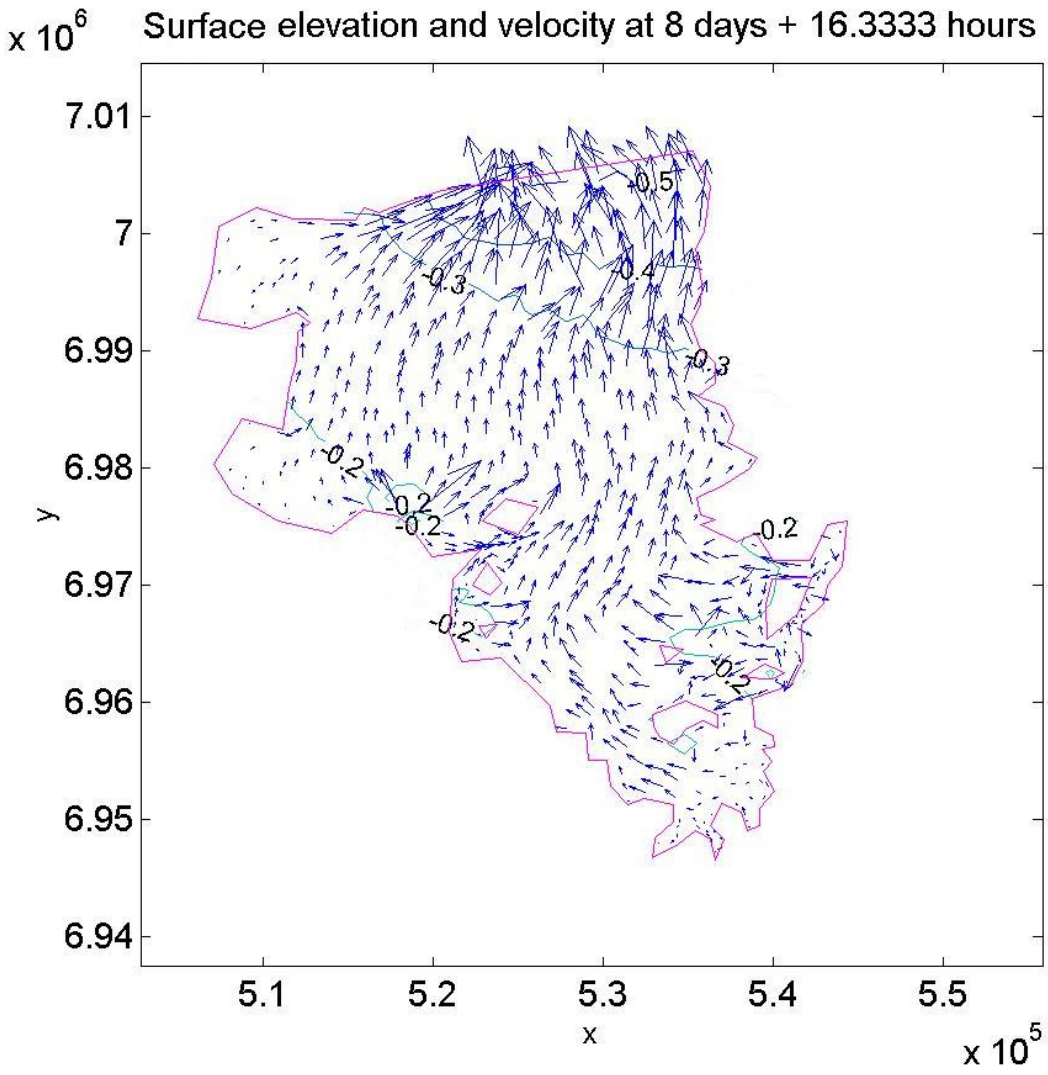


Figure 7.7: e) Surface elevation and velocity plots for periodic boundary conditions on a domain of Moreton Bay (8 days and 16 hours, mid-ebb).

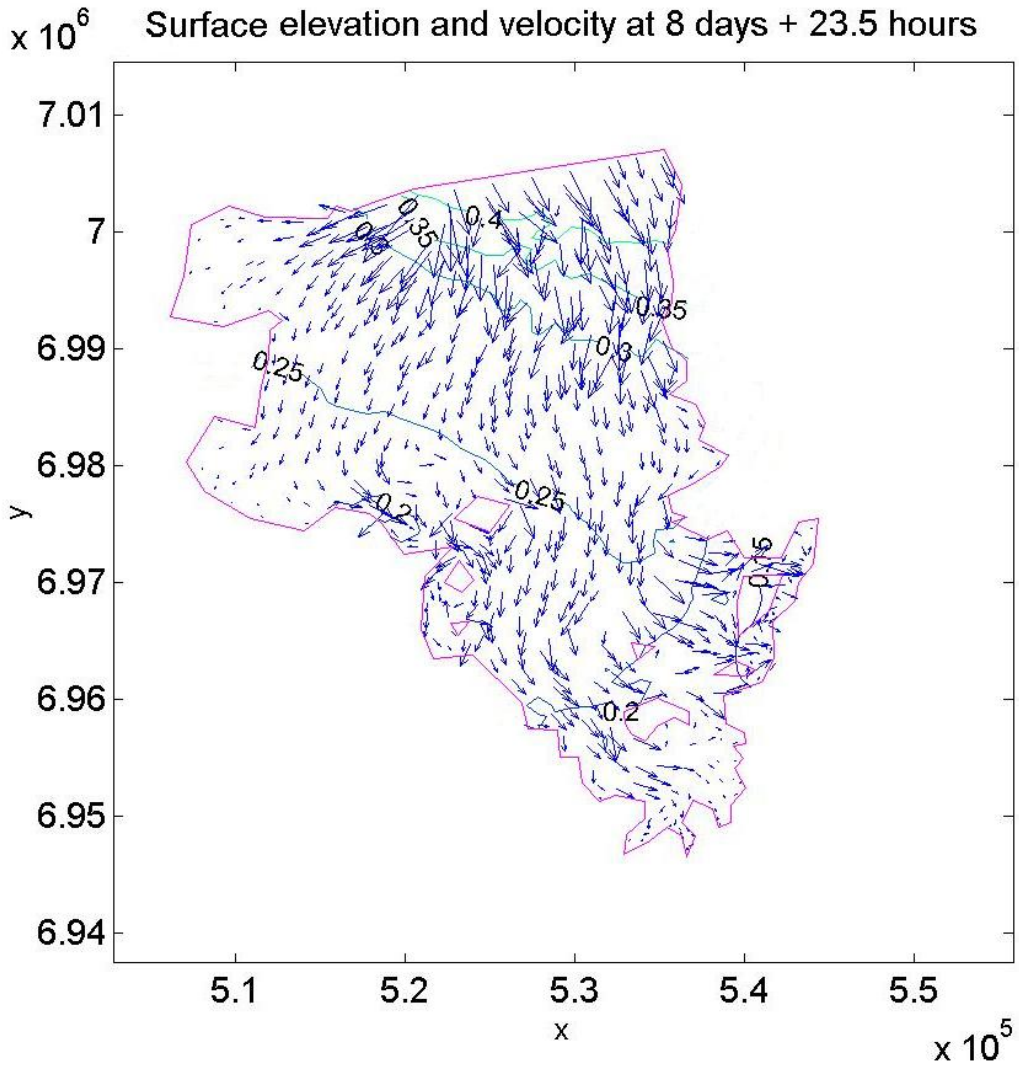


Figure 7.7: f) Surface elevation and velocity plots for periodic boundary conditions on a domain of Moreton Bay (8 days and 23.5 hours, mid-flood).

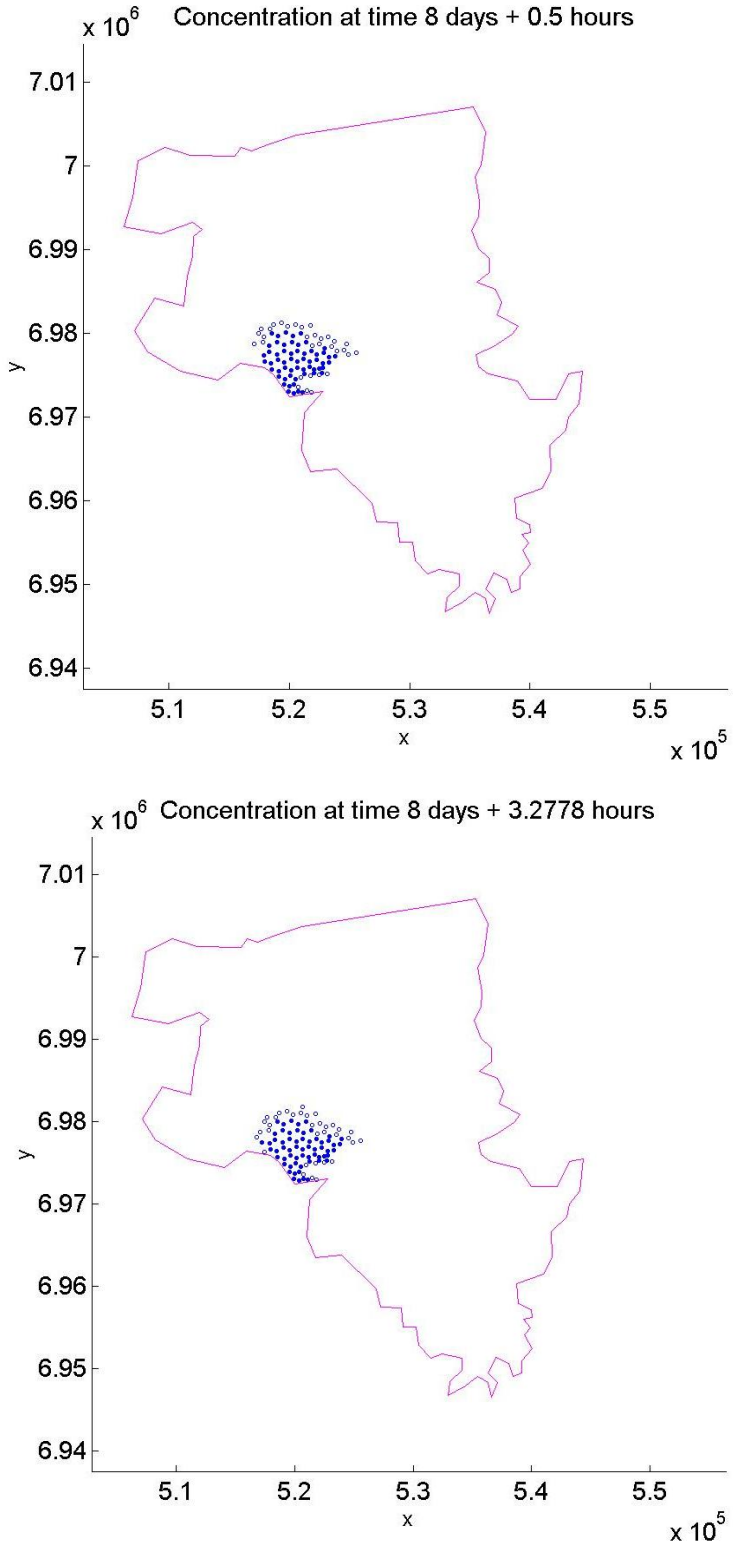


Figure 7.8: a) Scatter plots of contaminant concentration with periodic ocean boundary conditions on a domain of Moreton Bay (0.5, 3 hours).

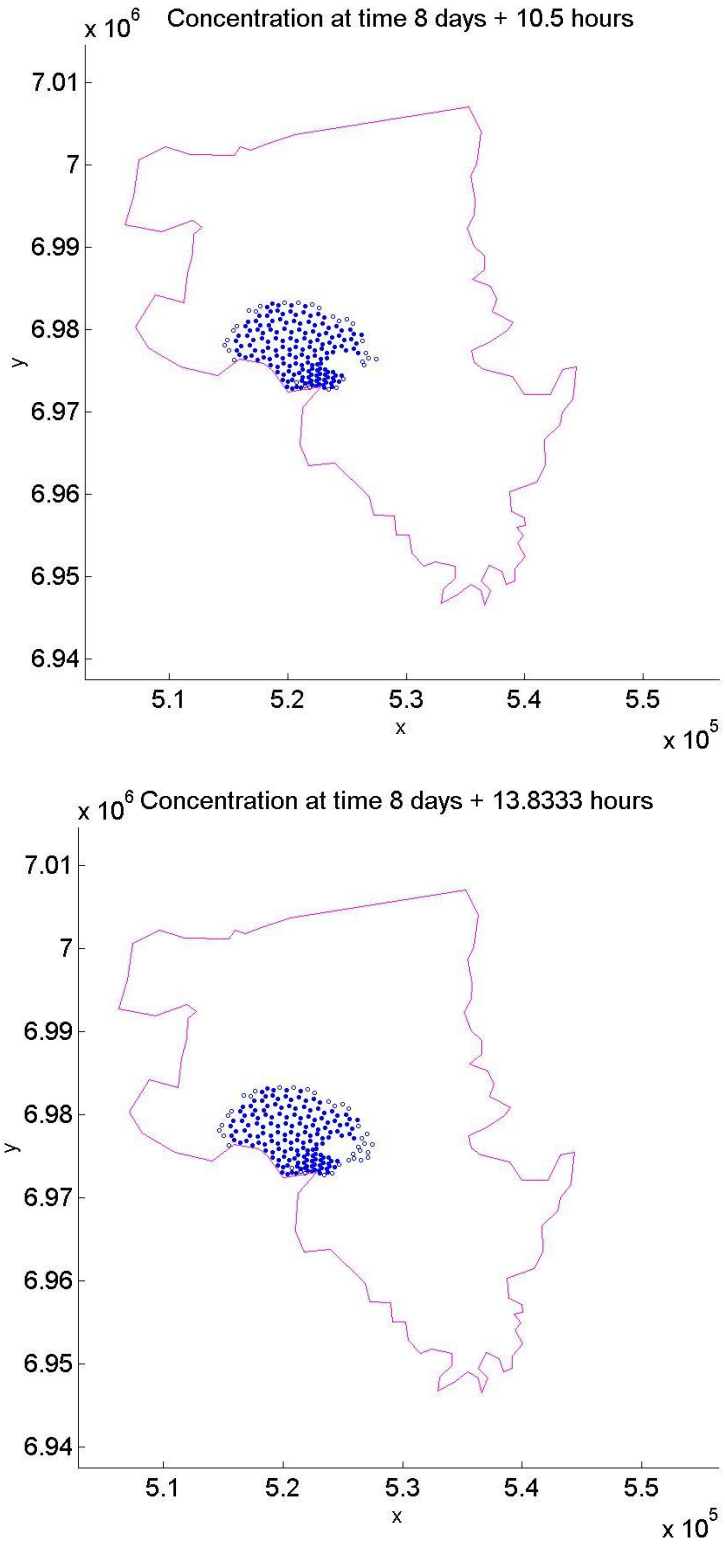


Figure 7.8: b) Scatter plots of contaminant concentration with periodic ocean boundary conditions on a domain of Moreton Bay (10.5, 13.8 hours).

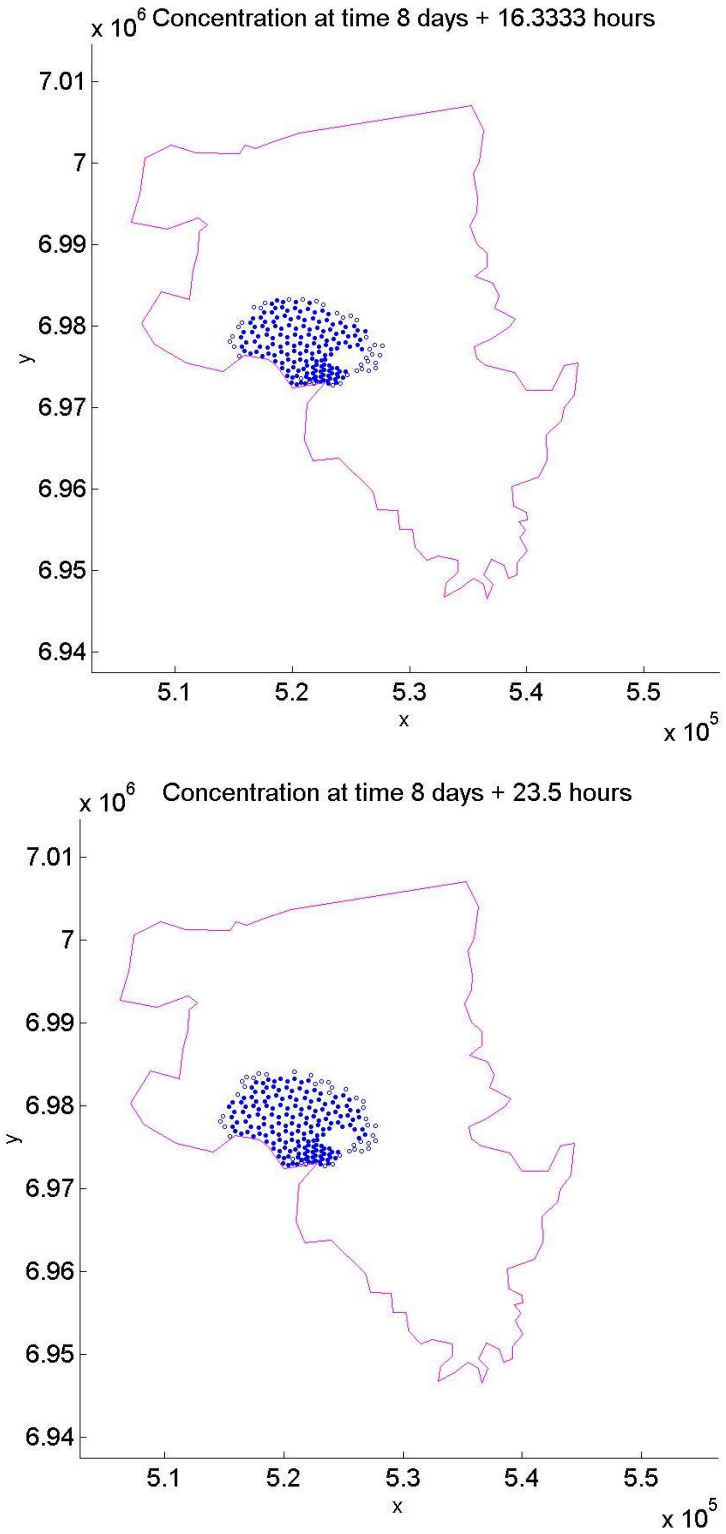


Figure 7.8: c) Scatter plots of contaminant concentration with periodic ocean boundary conditions on a domain of Moreton Bay (16, 23.5 hours).

7.4.2 Effect of wind on pollutants

It has been shown by Whitmore and DeLacy (2004), Costanzo *et al.* (2005), Costanzo *et al.* (2001) that Bramble bay, the area north of the Brisbane river, has higher levels of pollution than elsewhere in Moreton Bay. Costanzo *et al.* (2001) introduce the method of using the stable isotopes of nitrogen and outline how the method can be used. This technique was then used (Costanzo *et al.* (2005)) to investigate the transport paths of nutrients by mapping where the stable isotopes travel. Measurements taken in the whole of the western part of Moreton Bay during four years show the mouth of the Brisbane river and Bramble bay to have the highest concentration of δ^{15}N isotopes (reproduced in Figure 7.9). In a publication for Brisbane City Council, Whitmore and DeLacy (2004) outline the importance of taking care of the environment in order to preserve tourism in the region, which is Australia's largest employment sector. The authors state that (page 51) Bramble bay "displays some of the poorest water quality in Moreton Bay", which is attributed to the high levels of nutrient-rich water flow from the Brisbane river, delivered via the particular topography surrounding Bramble bay. The Healthy Waterways water monitoring organisation in Australia published a report card of water quality in Moreton Bay in 2009 [1], in which it awarded Bramble bay the worst possible mark of any part of the Bay, an F. The overall mark given for the whole Bay is a B, which corresponds to good quality.

The effect of wind on the path of a contaminant was tested by repeating the previous test with a value of minDepth = 0.5 m under the following different wind configurations:

- The average speed and direction from January to September is 3 ms^{-1}, in the southeasterly, southerly, or southwesterly directions
- From October to December, the wind direction is arbitrary and storms are more frequent. We choose a speed of 7 ms^{-1} under northerly and easterly directions.

This information comes from the Australian Bureau of Meteorology [2] and is summarised by the wind roses in Figure 7.10. The wind speeds are 35-year averages taken between 1951 and 1986 and the wind directions are based on observations between 1950 and 2000.

1. `www.health-e-waterways.org/reportcard/2009/Moreton%20Bay`
2. Wind roses can be found at `www.bom.gov.au/climate/averages/wind/selection_map.shtml` and average wind speeds can be found at `www.bom.gov.au/climate/averages/`

Figures 7.11 to 7.15 show the results of these simulations and are plotted so that elements with concentration $c > 0.05$ are filled circles and elements with concentration $0.001 < c < 0.05$ are empty circles. Elements with concentration less than $0.001\,\mathrm{kg\ m^{-3}}$ are not shown. The figure shown is at the second high tide after 8 days for the following tests. The figures for earlier times can be found on the CD at the back of this thesis.

Figures 7.11, 7.12 and 7.13 show a contaminant released under the conditions of a $3\,\mathrm{ms^{-1}}$ southeasterly, southwesterly and southerly wind, respectively. The contaminant appears to be very little affected by the wind, advancing towards Bramble bay only slightly more than with no wind. The discharge from Brisbane river has a greater effect than the wind because the water depth here is relatively deep (around 10 m) and uniform. There is little difference between the three wind directions, although the southeasterly seems to push the contaminant further into Bramble bay.

Figure 7.14 shows a contaminant released under the conditions of a $7\,\mathrm{ms^{-1}}$ northerly wind. Even under this relatively strong wind, the spread of contaminant is not very different from the conditions with no wind, although it does travel more to the south and west of the point of release because of the direction of flow.

Figure 7.15 shows a contaminant released under the conditions of a $7\,\mathrm{ms^{-1}}$ easterly wind. The contaminant is prevented from spreading so far west as in the other cases. This is partly due to the strength of the wind but also because the water is shallower in Bramble bay, resulting in a higher level of transport due to the wind shear.

The results show that it is clear that both the wind and the Brisbane river have a strong effect on the movement of a pollutant in this Bay. It is shown that the contaminant is moved towards Bramble bay under any wind direction, in agreement with field measurements (Costanzo *et al.* (2005)). Where the bathymetry is shallow, the effect of wind is greater, which agrees with the findings of Chapter 6 and Butman *et al.* (2008), Neill *et al.* (2008) and You (2005b).

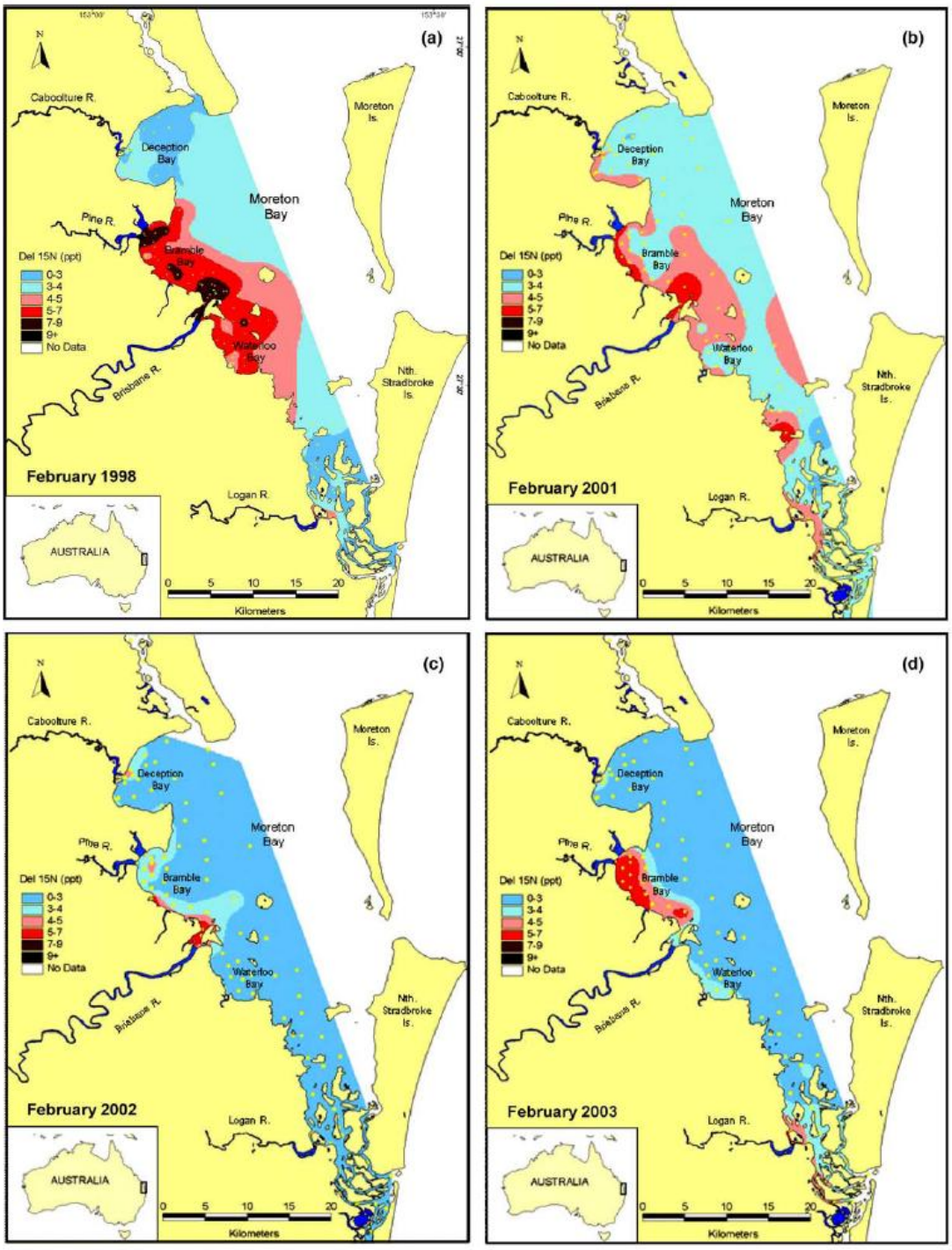


Figure 7.9: Spatial distribution of deployed macroalgal δ^{15}N values in February (a) 1998; (b) 2001; (c) 2002 and (d) 2003. Macroalgae (Catenella nipae, Rhodophyte) was deployed at approx. 100 sites (yellow solid circles) in Moreton Bay, Australia. Reproduced from Costanzo *et al.* (2005).

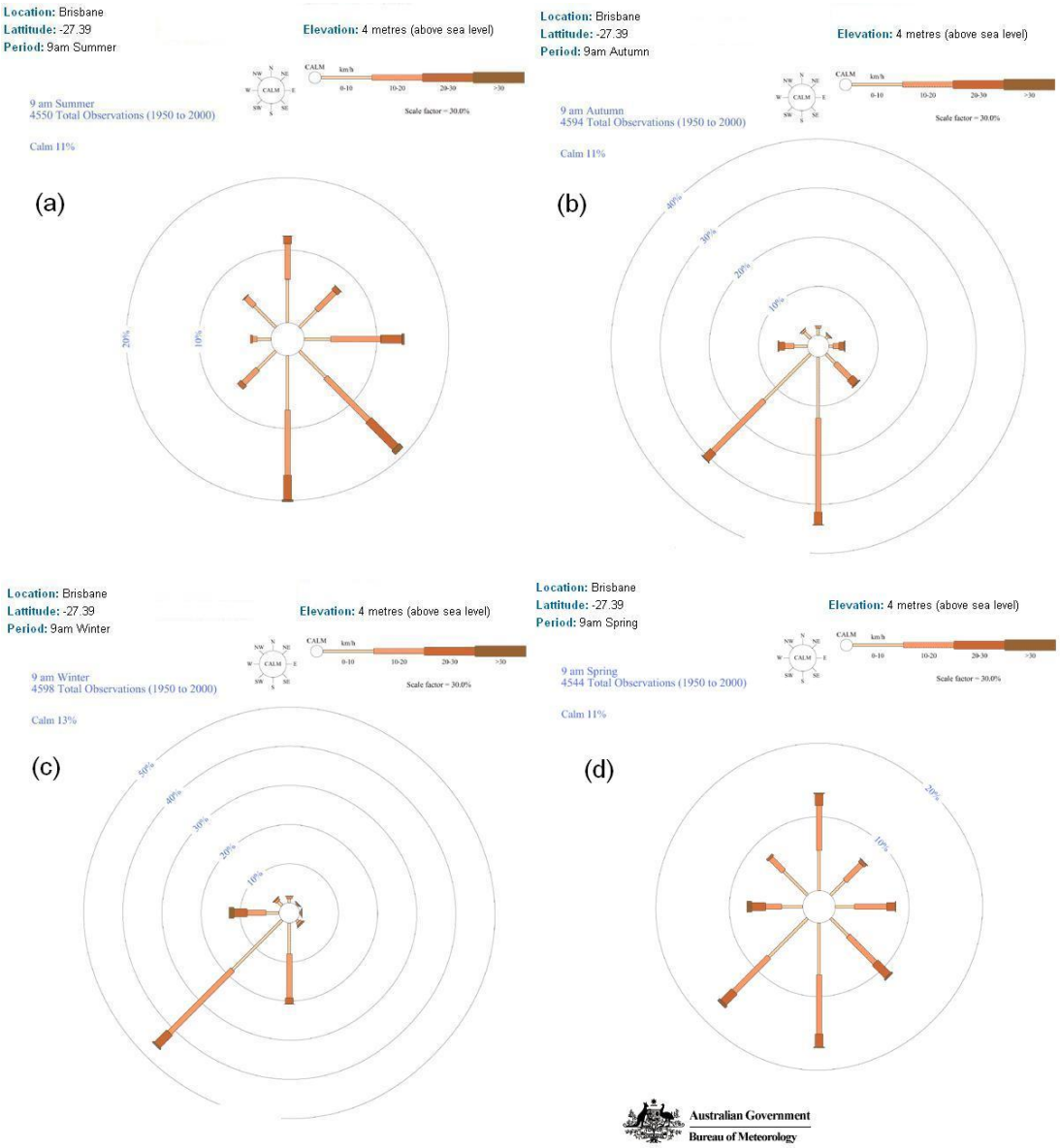


Figure 7.10: Wind roses (speed and direction) for summer (a), autumn (b), winter (c) and spring (d) in Moreton Bay. Reproduced from the Bureau of Meteorology, Australia.

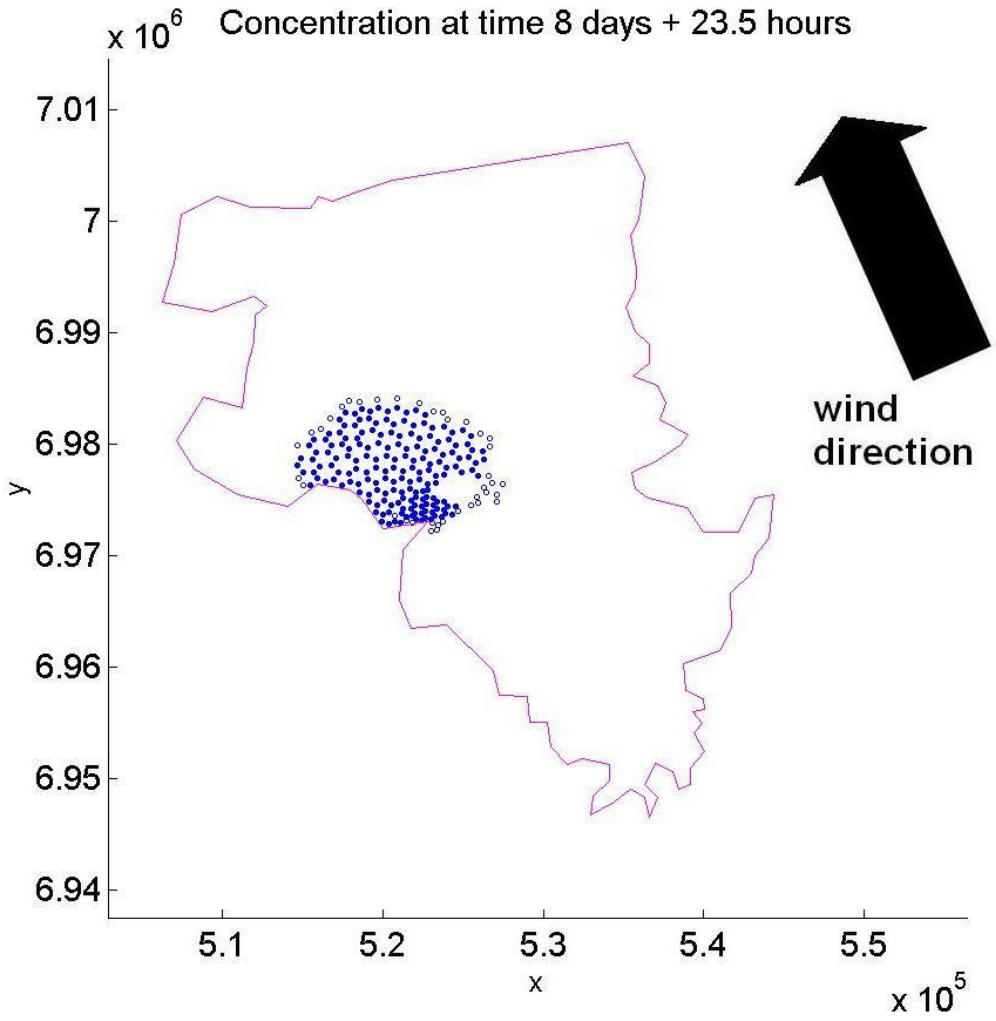


Figure 7.11: Scatter plots of contaminant with tidal ocean boundary conditions and a south-easterly wind of $3\,\mathrm{ms}^{-1}$ (23.5 hours).

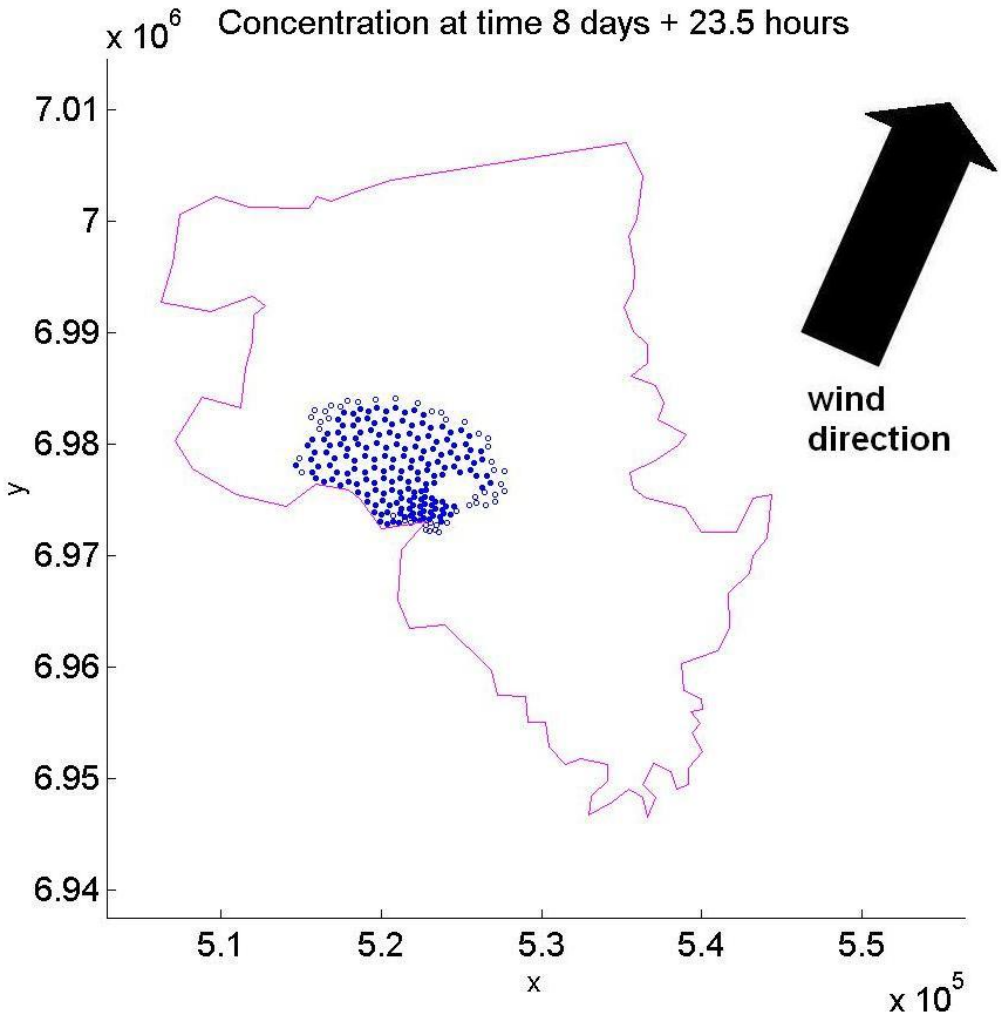


Figure 7.12: Scatter plots of contaminant with tidal ocean boundary conditions and a south-westerly wind of $3\,\mathrm{ms}^{-1}$ (23.5 hours).

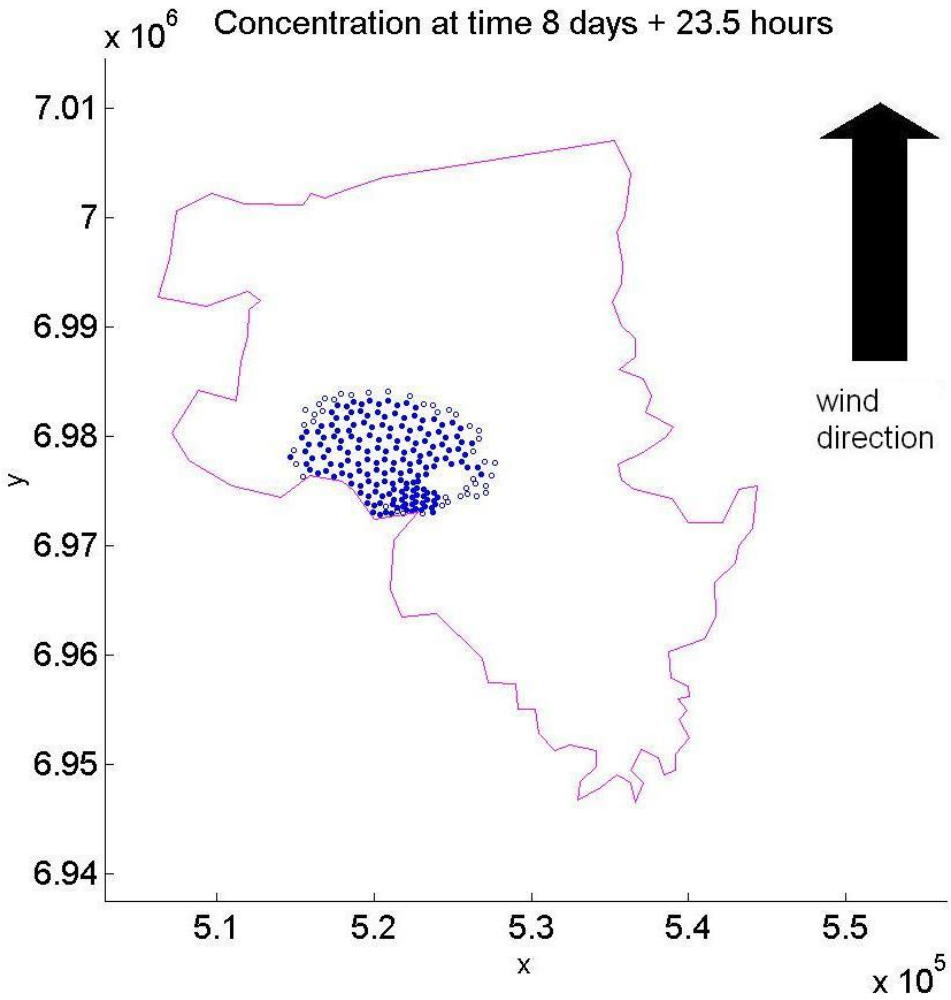


Figure 7.13: Scatter plots of contaminant with tidal ocean boundary conditions and a southerly wind of $3\,\mathrm{ms}^{-1}$ (23.5 hours).

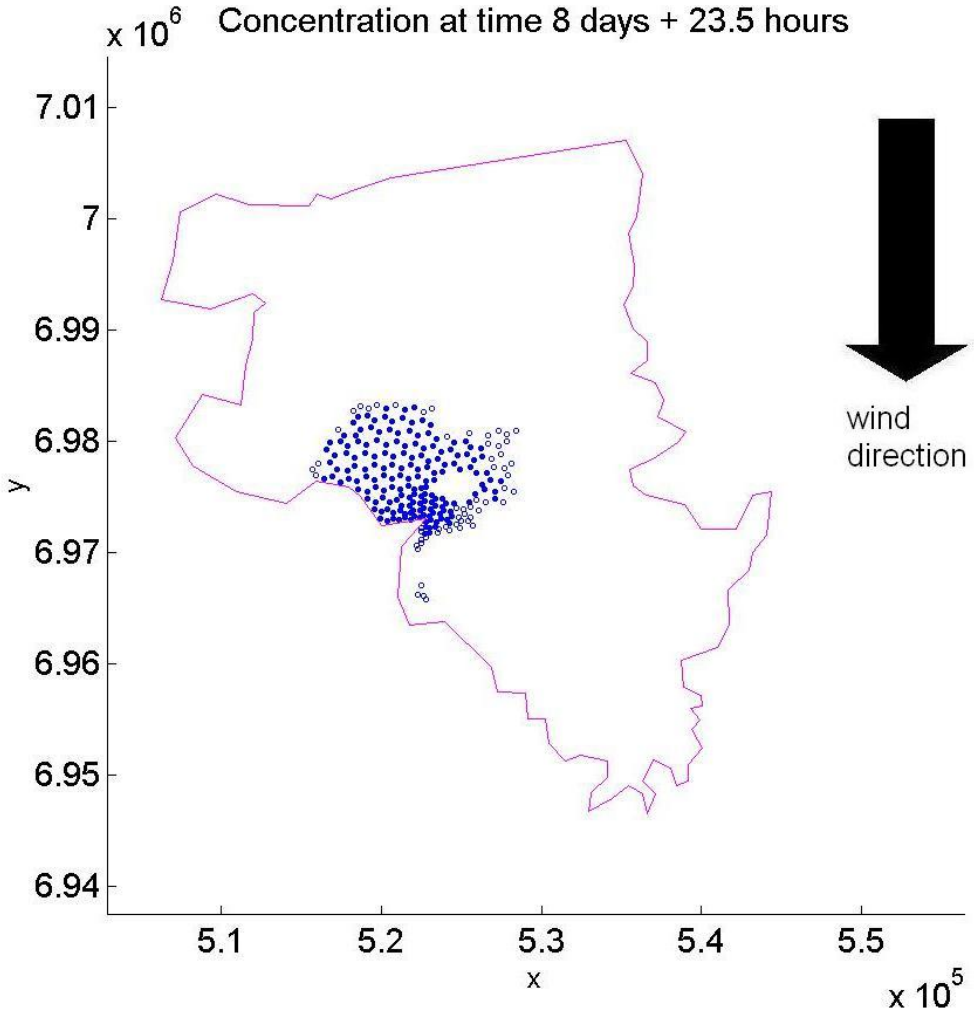


Figure 7.14: Scatter plots of contaminant with tidal ocean boundary conditions and a northerly wind of $7\,\mathrm{ms}^{-1}$ (23.5 hours).

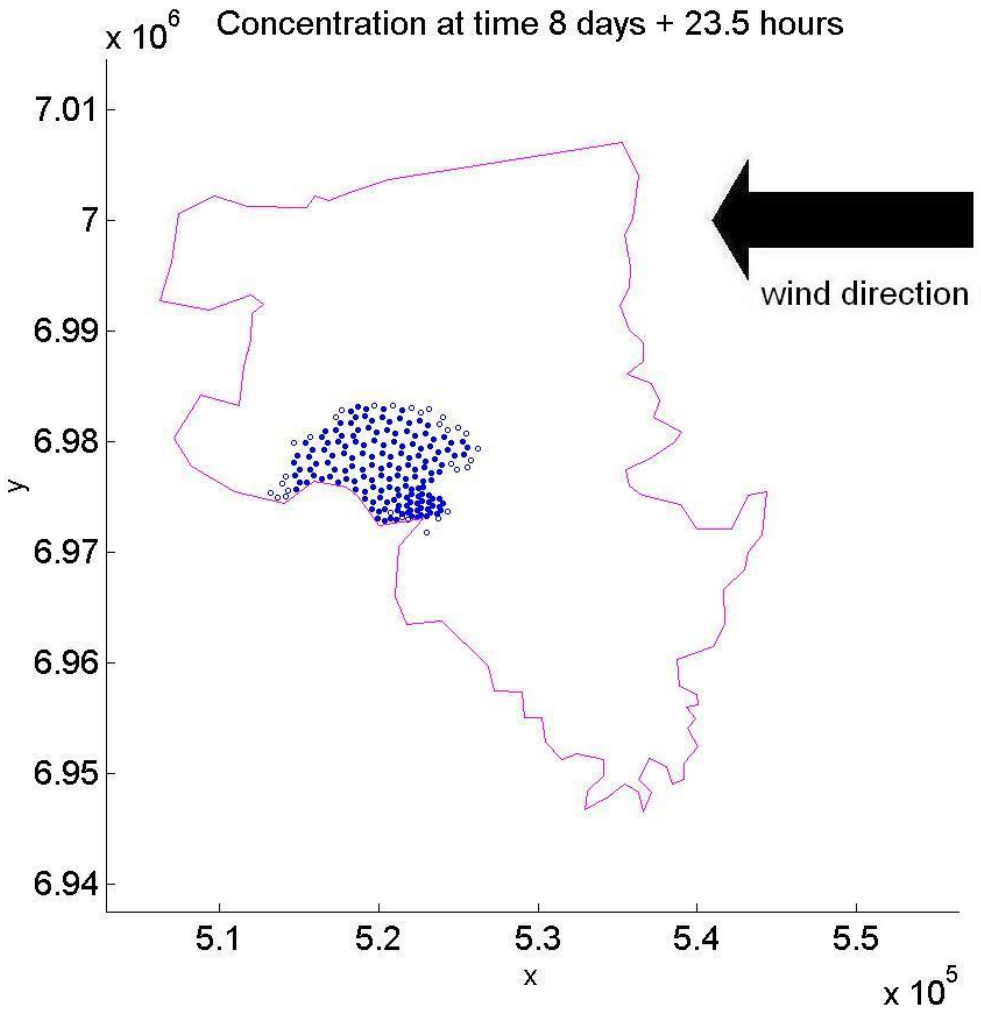


Figure 7.15: Scatter plots of contaminant with tidal ocean boundary conditions and an easterly wind of $7\,\mathrm{ms}^{-1}$ (23.5 hours).

7.5 Summary

In this chapter we have

- *Discussed the importance of monitoring the amount of pollution that enters Moreton Bay from the Brisbane river and the potential threat of pollution on the ecosystem there.*
- *Used field data to derive an approximate function of surface elevations and flow current to reproduce tidal forces in Moreton Bay. Riemann2D reproduces these functions well under both prescribed and reflective flux conditions. This provides confidence in previous simulations where reflective flux was used due to lack of field data.*
- *Shown that the direction of the prevailing wind has little effect on the path of the contaminant in areas of deeper water because the discharge from the Brisbane river is stronger.*
- *Demonstrated that with an easterly wind, because contaminant is moved into shallower water, it spreads over a larger area. This agrees with work from Butman et al. (2008), You (2005b) and Costanzo et al. (2005).*

Results and tests from Chapters 5 and 6 mean we can interpret these results with confidence. We know that *Riemann2D* can represent physically realistic and numerically stable results for simple (flat or axisymmetric bathymetry) domains. We have been able to understand local elevation and current speed in Moreton Bay thanks to field data from You (2005b), with which our results match well. Thus *Riemann2D* has demonstrated a promising capability for the realistic modelling of two-dimensional flows within harbours and bays of varying bathymetry.

CHAPTER 8

Conclusions

8.1 Summary

We have modelled flows in the shallow water in two test domains and in the environment of Moreton Bay, Australia. To do this we have used the classical shallow water, or St Venant, equations and a numerical shock-capturing FVM scheme written in an object-oriented programming language.

Some of the main results have been

- A newly-implemented time-dependent boundary condition has been found to effectively reproduce tidal forces (Chapters 4 and 7).
- Areas of larger bedslope were shown to produce more kinetic energy and less predictable flow velocities (Chapter 6).
- The path of a pollutant has been found to be much more strongly influenced by the wind than by the tide in domains where bed friction is important. The result is still true, although less strong, for small bedslope gradient. It is carried further in areas of higher bed friction (Chapter 6).
- Systematic testing of *Riemann2D* has shown it to be able to reproduce known analytical results (Chapter 5). We have also shown that *Riemann2D* converges and from Vázquez-

Cendón (1999) we know the scheme is first order accurate (higher accuracy is achieved with the MLG limiter). The C-property has also been established.

- Moreton Bay tidal simulations results have been successfully calibrated (Chapter 7).
- Simulations of pollutant transport in Moreton Bay have been shown to agree well with field observations (Chapter 7).

8.1.1 FVM and Java

Riemann2D was adapted and extended for this thesis. The implementation of a time-dependent boundary condition allowed the successful simulation of tides. The boundary condition was tested on some idealised domains, and demonstrated the correct behaviour, being periodic in nature and having realistic tidal range. Using a curve of best fit based on measurements from Moreton Bay as the boundary condition, simulation of flow in the Bay was found to give good agreement.

Riemann2D 's shock-capturing abilities were not exploited in this thesis. Instead it has been demonstrated that the code that was originally developed for small domains and fast flows also gives valid results for a large domain with slow flows and can effectively model pollutant transport over long time.

8.1.2 Analysis

The analysis of Chapter 3 demonstrated some of the basic properties of hyperbolic, shallow water equations to help understand the nature of the system. The results of the asymptotic analysis are part of the validation process.

Asymptotics

The asymptotic analysis that was carried out suggested that on a domain of constant depth of 10 m, to two orders of magnitude, the surface elevation is modelled by the depth of still water plus one tenth of the tidal component, *i.e.* $h = -b + \varepsilon\phi(t) + \mathcal{O}(\varepsilon^2)$. This provided insight into what to expect from the numerical simulations of this domain.

Source terms

Asymptotic analysis of the source terms showed that increasing wind speed by one order led to changes in the solution of two orders. The wind term was the largest source term we considered and had the largest effect on the solution. Increasing the bed friction term so that Manning's n is almost as large as physically probable led to corrections at $\mathcal{O}(\varepsilon^4)$ or smaller. In a domain of constant depth, the bed friction term adds very little to the flow.

These results were used to compare with the numerical results.

8.1.3 Model validation

Systematic validation has shown *Riemann2D* to reproduce the correct behaviour.

Riemann2D was validated using idealised bays: a square with flat bottom topography of side length 100 km, depth 10 m; and a circular basin of 45 km radius and maximum depth 10 m. The size of the source terms was varied to compare with asymptotics and some analytical results were reproduced.

Parameter variation for a square

In Chapter 5 it was seen that for an increase of one order of magnitude of wind speed there was a corresponding increase in the average flux by two orders of magnitude. The stronger the wind, the more energy is transferred to the water in the form of flow velocity. This agreed exactly with the asymptotic results from Chapter 3.

It was found that with an increase in the bed friction coefficient, the energy gained by the system decreased. Flow velocities were lower, as a higher roughness coefficient means energy is dissipated and not maintained in the flow. The results showed a change in the flux in the fourth decimal place, agreeing with the asymptotics.

Parameter variation for a circular basin

Results for the square case study showed changes that were whole orders of magnitudes, because of its uniform bathymetry. For the circular basin, however, it was found that both increasing wind speed and decreasing the bed friction coefficient have more effect in adding energy to the system (than for the case of the square). This was due to the vanishing bathymetry found in

the circular basin, as shallower regions are more subject to friction effects, which results in the generation of more kinetic energy. The changes in the average flux did not agree exactly with the asymptotics, but were consistent and followed the expected pattern, changing the solution by between one and three orders of magnitude.

Analytical results for a circular basin

Riemann2D was used to simulate solid-body rotation in the circular basin-like domain to compare with the analytical results of that problem. Without a numerical limiter, *Riemann2D* showed fairly poor agreement and there was visible smearing of the solution at the boundary of the domain. The tests were repeated using the MLG limiter and numerical solutions matched the exact solution very well, even for a low mesh density. *Riemann2D* could therefore be expected to preserve the correct solution, and the MLG limiter would be used in all further tests.

The solid-body rotation tests were repeated on meshes of three different densities, and by considering the root mean square value of the analytical solution from the simulated one, we saw that the computed solution approached the analytical as the mesh was refined, *i.e.* the scheme converged.

A steady-state wind blowing over a smaller circular basin was reproduced using *Riemann2D* and showed excellent agreement with the numerical results of Borthwick *et al.* (2001a) and Borthwick *et al.* (2001b) and the analytical prediction of Kranenburg (1992). This test was repeated on the larger domain in Chapter 6 and the same flow profile was reproduced.

8.1.4 Time-dependent boundary condition

The same two idealised bays, a square and a circular basin, were used to test the time-dependent boundary condition and to see the effect on the transport of a pollutant of wind and tide (Chapter 6).

Tide

The periodic nature of the tidal boundary condition was seen in the similarity of surface elevation and velocity profiles at 6-hour intervals. This corresponds to a semi-diurnal tide, as is found on the Queensland coast. The tidal range was seen to be 0.9 m in the square, a typical

value for that coastline. The range was smaller for the circular basin, which is attributed to the relatively narrow and shallow ocean boundary through which a limited volume can flow. An approximation of the tide was made by using just the M_2 component, which nevertheless gave realistic results.

The simulations were all made for at least 10 days to ensure that any effects from the initial steady-state were removed. All the tests produced the correct behaviour.

Coriolis force

It was found that Coriolis force was strong enough to disrupt the otherwise symmetric nature of the flow, even though it did not appear in the first three terms of the asymptotic solution so is small. Its effect was weaker in the circular basin, as bed friction effects are stronger here.

In the circular basin with wind, the two gyres of recirculation found in Chapter 5 were recreated. The difference was due to the Coriolis force that meant the dividing line between the two gyres was no longer straight, but twisted due to the spin of the Earth.

Pollutant transport

It was found that tidal forcing alone does not greatly affect the transport of a pollutant in areas where bed slope is zero. Wind was found to be much more important, especially in the case of the circle where the shearing effects from the wind, combined with shallower bathymetry, led to higher levels of transport. This agrees with Butman *et al.* (2008), Neill *et al.* (2008) and You (2005b). In particular, areas of high bed friction correspond to the areas where the pollutant is carried to. Pollutant was influenced by Coriolis to the extent that it is carried with the flow which is itself affected by Coriolis.

8.1.5 Moreton Bay

Riemann2D showed good agreement with measurements when those (best fit for) tide heights and tidal currents measured by You (2005b) were used as boundary conditions on the Moreton Bay domain (Chapter 7). The simulation where the flux boundary condition was set as transmissive was also shown to be realistic. Although the present function gave very good agreement, it was taken from data measured about 20 km from the ocean boundary, and it

would be preferable to use data that were collected from the ocean boundary itself. It is not known what wind speeds or directions were present during the measurements, nor what effect they would have on the flow. More precise measurements could improve the present results, which are already good, and which are as accurate as those in the literature.

Simulations on the Moreton Bay domain with no wind, the average wind speed and three different directions, and a moderate breeze in two different directions were carried out. It was found that the contaminant released at the mouth of the Brisbane river was transported towards Bramble bay, to the northwest of the river. This agrees with the literature, that this is the most polluted part of Moreton Bay. The wind had only a little effect on the path of the contaminant.

Under a moderate breeze in the easterly direction, contaminant was moved furthest towards Bramble bay, whereas the northerly wind tended to take the contaminant more to the south of the Brisbane river and since this part of the Bay had deeper bathymetry, contaminant was contained in a smaller area.

We have made no attempt to conclude what level of contaminant would be harmful to the environment of Moreton Bay. The results we have obtained are general, and can be used to predict the qualitative behaviour of a contaminant release. The initial concentration value of $0.3\,\mathrm{kg\ m^{-3}}$ was chosen as a purely illustrative value.

Flow in the southern part of the Bay could not be considered to be realistic due to the large number of islands there. The boundary conditions on the islands were reflective and meant flow velocities could be artificially high. Realistically they should be allowed to flood and dry. No tidal condition was set on the southern passage but since this is an area of high friction, tidal effects would be limited (Larsen (2007)) and some preliminary tests have suggested this would lead to numerical instabilities.

8.2 Further work

It is valuable to have a mathematical model as a way of understanding the main processes that affect water flow in a bay. A more sophisticated and interdisciplinary model could be developed that includes management strategies by taking advantage of the object-oriented structure of the *Riemann2D* code.

8.2.1 Boundary conditions

Data for tidal currents and heights measured at the ocean boundary would allow a more accurate model for the boundary conditions. Although a sine curve fitted to observed data gave good results, it would be ideal to modify the code so that measured data may be read from a file.

8.2.2 Time-dependent wind

The speed of the wind that is measured in a real bay is not constant over several days, or even over several hours. In the case of Moreton Bay, the wind speed is consistently higher at 3pm than at 9am. Although there is currently no way to model this in *Riemann2D* , it is certainly an area that needs to be improved in order to obtain a more realistic simulation.

8.2.3 Islands

There are a number of islands in the southern part of Moreton Bay, that in this model had reflective boundary conditions. Ideally, the islands would be allowed to flood and dry as the tide comes in and goes out. A new version of *Riemann2D* is designed to allow flooding and drying, although there has not been time to test and use the new version in this thesis. It would be interesting to do this, and allow flow in the southern part of the Bay to be treated more realistically.

8.2.4 Delta region

At the southern end of Moreton Bay there is a large number of small rivers and tributaries, forming a delta region. In the numerical simulations we have put reflective boundaries on this end, but, as in the case of the islands, it would be preferable to represent the fact that water flows in and out of this region. We might also improve the model of Brisbane river discharging water into the Bay by making it vary in time.

A more realistic localised model of Brisbane river would also be useful for modelling the path of a pollutant under a north wind, as at the moment it is unclear whether it flows back into the Brisbane river or not.

8.2.5 Contaminant diffusion

The advective concentration equation could be made into an advection-diffusion equation by the modification of the right hand side with relevant diffusion coefficients. Data are needed to estimate these coefficients, and if this were available, would provide improved accuracy in predicting the fate of a pollutant.

8.2.6 Stratified model

The assumption of averaging over the vertical axis and assuming motion only in the horizontal plane is arguably unrealistic. In reality, the effect of bed friction on the bottom of the bed causes the water to flow in a different way from the surface water that is more strongly subject to the effects of wind. Thus a column of water does not behave in a uniform way, even though this is a fair assumption for a mathematical model. It is possible to use the full 3D Navier-Stokes equations, but as a simpler approach, we could assume a stratified model separated by a viscous boundary, with the lower layer subject to bed friction and the upper layer affected more by the wind. Some analysis to this problem was given in Chapter 3.

8.2.7 Erosion model

A 2D erosion model based on the (1D) work of Cao *et al.* (2004) has been developed for *Riemann2D*. Early tests show good agreement with the 1D results on a two-dimensional dam-break problem, although the results for slow flows and natural boundaries are less realistic (more erosion happening in deeper parts of the bay than shallower). If the erosion extension can be improved, it will provide valuable insight into the areas of Moreton Bay that are most subject to sediment transport. In practice, this will be a complex task as different areas of Moreton Bay are composed of different types of sediment: there is a large sandy area in the central-northern part, while areas to the south have finer clay or silt which is sticky and not easily entrained. Thus a realistic sediment transport module would need to be able to handle a distribution of sediment size classes rather than a typical single representative size class.

8.3 Conclusions

Riemann2D has been systematically validated on idealised domains and for various flow conditions and has shown consistent and reliable results. The domains it has been tested on are much larger than those it was designed for. The problem modelling times have also been much longer. The code exhibits the correct behaviour and shows good agreement with analytical results. The tide-like boundary condition has been applied in the model of flow in a real bay and results calibrate well with field measurements.

Modelling the path of a contaminant in Moreton Bay has successfully highlighted Bramble bay as being at the greatest risk of pollution, and more so under a moderate (or stronger) easterly wind. This information could be used in management decisions in how to protect the environment of Moreton Bay.

APPENDIX A

Derivation of the model

We consider a large harbour of shallow water, of typical dimensions 100km across, and 10m deep. At one end of the harbour there is a tide (via a connection with the ocean or sea) that changes the level of the water in the harbour by a small proportion – typically 1-2 metres over a 12 hour period. At first we consider the one dimensional model, where all variables are functions of horizontal distance x and time t, and then later the two dimensional.

Let the geometry of the problem be as described by Figure 2.1 in Section 2.1.

A.1 Mass conservation: derivation

The classical shallow water equations (SWEs) can be derived in a number of ways: from taking the three-dimensional Euler equations and making a hydrostatic approximation; from first principles by considering a block of fluid as it travels horizontally around within the domain; or by considering the forces acting on a block of fluid with sides of infinitely small width. In all cases the SWEs represent that mass and momentum of the fluid are conserved.

Conservation of mass (Euler)

This derivation is the classical Eulerian approach, which involves studying a block of particles as it moves from a *fixed reference point.*

Consider the height of the water in a horizontal cross-section from x to $x+\delta x$ measured at mean sea level. The height is denoted $h = h(x,t)$ so we can write an expression for the area of the cross-section, by integrating from x to $x+\delta x$, as

$$\int_x^{x+\delta x} h(\hat{x},t)\,\mathrm{d}\hat{x} = \text{volume of water under } h(x,t) \text{ between } x \text{ and } x+\delta x \text{ per unit width.}$$

The rate at which this volume changes over time is

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_x^{x+\delta x} h(\hat{x},t)\,\mathrm{d}\hat{x}, \tag{A.1}$$

which has dimensions [$\mathrm{m}^2\ \mathrm{s}^{-1}$] per unit width. This expression, when there are no sources, represents the net amounts of fluid flowing past $(x,\ x+\delta x)$ per unit time. But the product $u(x,t)h(x,t)$ (the x-flux per unit width) also represents the amount of water flowing past the point x. So (A.1) is equal to the change in the flux between x and $x+\delta x$. More formally,

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_x^{x+\delta x} h(x,t)\,\mathrm{d}x &= u(x,t)h(x,t) - u(x+\delta x,t)h(x+\delta x,t)\\ &=: -u(\hat{x},t)h(\hat{x},t)\Big|_x^{x+\delta x}.\end{aligned}$$

Assuming that the functions $u(x,t)$, $h(x,t)$ are smooth (*i.e.* continuously differentiable, and do not model jumps or shocks in the system) we can change the order of integration and differentiation by Leibniz' rule on the left hand side, and rewrite the right hand side as an integral, to get

$$\begin{aligned}\int_x^{x+\delta x}\frac{\partial}{\partial t}(h(\hat{x},t))\,\mathrm{d}\hat{x} + \int_x^{x+\delta x}\frac{\partial}{\partial \hat{x}}\left(u(\hat{x},t)h(\hat{x},t)\right)\,\mathrm{d}\hat{x} = 0\\ \Rightarrow\ \frac{\partial h}{\partial t} + \frac{\partial\,(uh)}{\partial x} = 0.\end{aligned}$$

This is the equation for conservation of mass in one spatial dimension. It can be extended to 2D by considering the same situation in the y-plane, where the flux is $v(x,y,t)h(x,y,t)$. Then the rate of change of the volume of water over time (A.1) in two spatial dimensions is

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_x^{x+\delta x}\int_y^{y+\delta y} h(\hat{x},\hat{y},t)\,\mathrm{d}\hat{x}\mathrm{d}\hat{y},$$

which is related to flux in the following way:

$$\begin{aligned}\int_x^{x+\delta x}\int_y^{y+\delta y}\frac{\partial}{\partial t}h(\hat{x},\hat{y},t)\,\mathrm{d}\hat{x}\mathrm{d}\hat{y} + \int_x^{x+\delta x}\int_y^{y+\delta y}\frac{\partial}{\partial \hat{x}}\left(u(\hat{x},y,t)h(\hat{x},y,t)\right)\,\mathrm{d}\hat{x}\mathrm{d}\hat{y} +\\ \int_x^{x+\delta x}\int_y^{y+\delta y}\frac{\partial}{\partial \hat{y}}\left(v(x,\hat{y},t)h(x,\hat{y},t)\right)\,\mathrm{d}\hat{x}\mathrm{d}\hat{y} = 0.\end{aligned}$$

On taking the limits as $\delta x \to 0$ and $\delta y \to 0$ (again, for the integrand continuous), we arrive at

$$\frac{\partial h}{\partial t} + \frac{\partial (uh)}{\partial x} + \frac{\partial (vh)}{\partial y} = 0.$$

In designing a computer program this idea is an important one to keep in mind: the equation will be discretised before it can be solved numerically, which means breaking down the domain into smaller pieces where the problem is solved and then 'stitched' back together. These pieces could be based on $(x,\, y)$, $(x + \delta x,\, y)$, $(x,\, y + \delta y)$, $(x + \delta x,\, y + \delta y)$ and the numerical method thought of as approximating the integrals.

Conservation of mass (The transport theorem)

A more sophisticated derivation makes use of the transport theorem, which is as follows: let an area $A(t)$ containing the same fluid particles as it moves around, with the horizontal velocity vector $\mathbf{u}(x, y, t)$

$$\frac{D}{Dt} \iint_{A(t)} F(\mathbf{x}, t)\, \mathrm{d}V = \iint_{A(t)} \frac{DF}{Dt} + F \nabla \cdot \mathbf{u}\, \mathrm{d}V,$$

for some function $F(\mathbf{x}, t)$, where the material derivative

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla,$$

and the vector $\mathbf{u} = [u, v]^{\mathrm{T}}$ is the velocity of the fluid.

The aim is to write down an equation which describes the fact that mass of fluid is conserved. Let the volume domain $A(t)$ be a column of water perpendicular to the (x, y) plane, of height h and small cross-sectional area $\partial A(t)$ at time t, which evolves to $\partial A(t + \delta t)$ at time $t + \delta t$. The column of water is moving with the flow, so in this sense the derivation can be considered to be following a Lagrangian co-ordinate system. We require that the water mass of the column is conserved, which is equivalent to saying that the rate at which the mass changes is zero:

$$\begin{pmatrix} \text{rate of change of mass in column} \\ \\ \text{that moves with the flow} \end{pmatrix} = \frac{D}{Dt} \iint_{A(t)} \rho h \, \mathrm{d}x \mathrm{d}y = 0, \tag{A.2}$$

where ρ is density of the substance.

Using the above transport theorem, (A.2) can be written

$$\frac{D}{Dt} \iint_{A(t)} \rho h \, \mathrm{d}A = \iint_{A(t)} \frac{\partial}{\partial t} (\rho h)\, \mathrm{d}A + \iint_{A(t)} \nabla \cdot (\rho h \mathbf{u})\, \mathrm{d}A = 0.$$

Using the divergence theorem, we can write the above as

$$\iint_{A(t)} \frac{\partial(\rho h)}{\partial t}\,\mathrm{d}A + \oint_{\partial A} \rho h\mathbf{u}\cdot\mathbf{n}\,\mathrm{d}S = 0.$$

Assuming continuity of the integrand, and that $A(t)$ is arbitrary, we can say that

$$\begin{aligned} \frac{\partial(\rho h)}{\partial t} + \nabla\cdot(\rho h\mathbf{u}) &= 0 \\ \Rightarrow \frac{\partial(\rho h)}{\partial t} + \frac{\partial(\rho hu)}{\partial x} + \frac{\partial(\rho hv)}{\partial y} &= 0, \end{aligned}$$

and as ρ is a constant it can be eliminated to give

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} = 0,$$

as required.

A.2 Conservation of momentum derivation

Momentum: x-direction

Consider a block of water of width δx in the x-direction by width δy in the y-direction, and height $h(x,y,t)$. Fluid enters or leaves at velocity $u(x,y,t)$ in the x-direction, and $v(x,y,t)$ in the y-direction, as shown in Figure A.1.

To derive the equation for conservation of momentum, we start with Newton's law of motion, $\mathbf{F} = m\mathbf{a}$, where $\mathbf{F}$ is the sum of all forces, m is mass, and $\mathbf{a}$ is acceleration of the block of water. Now consider what forces the control volume in Figure A.1 is subject to

i) **Change in momentum.** Momentum $= p =$ mass$\times$velocity, so let mass of the control volume be $\rho(u(x,y,t)+v(x,y,t))h(x,y,t)\delta t$, and let velocity be $u(x,y,t)$, then change in momentum, per unit of time δt is

$$\frac{\delta p}{\delta t} = \int_x^{x+\delta x} \frac{\partial}{\partial\hat{x}}(\rho u^2(\hat{x},\hat{y},t)h(\hat{x},\hat{y},t))\,\mathrm{d}\hat{x} + \int_y^{y+\delta y} \frac{\partial}{\partial\hat{y}}(\rho u(\hat{x},\hat{y},t)v(\hat{x},\hat{y},t)h(\hat{x},\hat{y},t))\,\mathrm{d}\hat{y}. \tag{A.3}$$

ii) **Force due to the free surface of the water.** This is related to the potential energy of the wave. The potential energy of the wave is $mg\eta = \rho h_*\delta x g(h+b)$, where $m = \rho h_*\delta x$ is mass, g is force due to gravity, $h_* = \frac{1}{2}(h(x,y,t)+h(x+\delta x,y,t))$ is the average height,

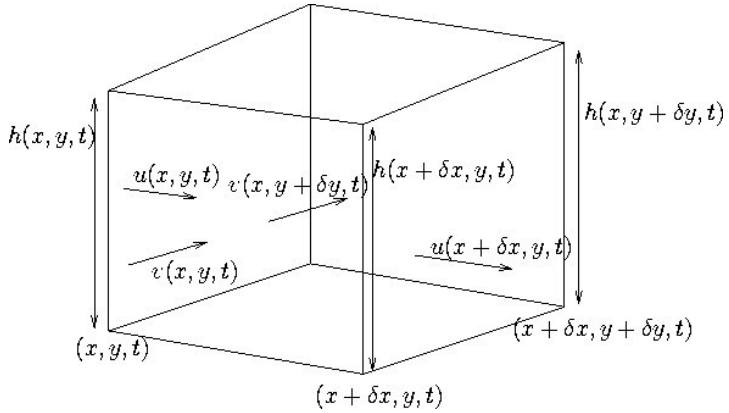


Figure A.1: A block of fluid showing the velocities and heights on the faces and edges of a typical control volume over the square $(x, x+\delta x) \times (y, y+\delta y)$.

and $\eta = h + b$ is the wave amplitude. Now, the change in energy over the control volume is due to changes in η, in other words, it is

$$\begin{aligned}\delta(PE) &= \rho g h_* \delta x\{(h(x+\delta x, y, t) + b(x+\delta x, y)) - (h(x,y,t) + b(x,y))\} \\ &= \rho g h_* \delta x\{(h(x+\delta x, y, t) - h(x,y,t)) + (b(x+\delta x, y) - b(x,y))\}.\end{aligned}$$

So the force due to the free surface of the water is change in energy per change in distance

$$\frac{\delta(PE)}{\delta x} = \int_x^{x+\delta x} \rho g h_* \frac{\partial}{\partial \hat{x}}(h(\hat{x}, y, t) + b(\hat{x}, y))\,\mathrm{d}\hat{x}. \tag{A.4}$$

iii) **The RHS of Newton's equation.** Total mass of the volume is $\rho \delta x h(x,y,t)$, and acceleration is the derivative w.r.t. time of $u(x,y,t)$. The RHS expression is therefore

$$\frac{\delta(m\mathbf{a})}{\delta x} = \frac{\mathrm{d}}{\mathrm{d}t}\int_x^{x+\delta x} \rho h(\hat{x}, y, t) u(\hat{x}, y, t)\,\mathrm{d}\hat{x} = \int_x^{x+\delta x} \frac{\partial}{\partial t}(\rho u(\hat{x}, y, t) h(\hat{x}, y, t))\,\mathrm{d}\hat{x}. \tag{A.5}$$

Combining the integrals (A.3), (A.4) and (A.5), dividing by ρ, and taking the limit as $\delta x \to 0$, we arrive at the equation of (A.6b)

$$\frac{\partial(hu)}{\partial t} + \frac{\partial(u^2 h + g h^2/2)}{\partial x} + \frac{\partial(huv)}{\partial y} = -gh\frac{\partial b}{\partial x}.$$

Momentum: y-direction

The derivation for the y-component uses the same arguments, except that we now interchange x with y and u with v. Alternatively, rotate the (x, y) axis by $\pi/2$. Note that in performing this rotation of axes, the new y axis will be pointing in the opposite direction from the direction of the original y axis, but since v was measured as positive in the original y axis-direction, we change the sign of *both* y and v so the equation remains invariant.

Making the same analysis as for the x-direction equation, we find the third governing equation, as stated in (A.6c):

$$\frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{\partial(hv^2 + gh^2/2)}{\partial y} = -gh\frac{\partial b}{\partial y}.$$

A.3 The governing equations

We have shown from first principles how mass and momentum are conserved in shallow water, hence derived the shallow water equations. In conservation form, the shallow water equations are

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} = 0; \tag{A.6a}$$

$$\frac{\partial(hu)}{\partial t} + \frac{\partial(hu^2 + gh^2/2)}{\partial x} + \frac{\partial(huv)}{\partial y} = -gh\frac{\partial b}{\partial x}; \tag{A.6b}$$

$$\frac{\partial(hv)}{\partial t} + \frac{\partial(huv)}{\partial x} + \frac{\partial(hv^2 + gh^2/2)}{\partial y} = -gh\frac{\partial b}{\partial y}. \tag{A.6c}$$

Because we can write these equations in vector form as

$$\frac{\partial}{\partial t}\begin{pmatrix} h \\ hu \\ hv \end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix} hu \\ hu^2 + g\frac{1}{2}h \\ huv \end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix} hv \\ huv \\ hv^2 + g\frac{1}{2}h \end{pmatrix} = \mathbf{s},$$

we call this a conservation form for the PDEs, and the conservation variables are $(h,\, hu,\, hv) = (q_1,\, q_2,\, q_3)$. Alternatively, we may write the above in non-conserved variables $(h,\, u,\, v)$ in vector form as

$$h_t + \mathbf{u} \cdot \nabla h + h\nabla \cdot \mathbf{u} = 0, \tag{A.7a}$$

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} + g\nabla h = \hat{\mathbf{s}}, \tag{A.7b}$$

where subscripts denote partial differentiation with respect to that variable and the vector $\hat{\mathbf{s}}$ is the vector of source terms in the momentum equations.

APPENDIX B

Mathematical analysis

B.1 Riemann invariants

Consider a general $n \times n$ system of the form

$$A\frac{\partial \mathbf{u}}{\partial x} + B\frac{\partial \mathbf{u}}{\partial y} = \mathbf{c}, \tag{B.1}$$

together with initial data $\mathbf{u} = \mathbf{u}_0(s)$, $x = x_0(s)$, $y = y_0(s)$ for some parameter $s_1 \leqslant s \leqslant s_2$. From the boundary data, $\dot{\mathbf{u}}_0 = \dot{x_0}\mathbf{u}_x + \dot{y_0}\mathbf{u}_y$ (a dot $\dot{}$ represents differentiation w.r.t. s, subscripts represent differentiation w.r.t. variables x or y). The partial derivatives of the system (B.1) are uniquely defined provided

$$\begin{vmatrix} A & B \\ I\dot{x_0} & I\dot{y_0} \end{vmatrix} \neq 0$$
$$\Rightarrow \ |B - \lambda_i A| \neq 0, \qquad \text{where} \quad \dot{y_0} = \lambda_i \dot{x_0}.$$

Characteristic curves are defined to be curves on which solutions for $\mathbf{u}_x$, $\mathbf{u}_y$ of (B.1) and the initial data may not be uniquely found, that is $|B - \lambda_i A| = 0$, where $\mathrm{d}y/\mathrm{d}x = \lambda_i$.

Our PDE (B.1) is said to be hyperbolic when there are real and distinct eigenvalues. The system (B.1) has a unique left eigenvector $\mathbf{l}_i^{\mathrm{T}}$ associated with each eigenvalue such that

$$\mathbf{l}_i^{\mathrm{T}}(B - \lambda_i A) = \mathbf{0}^{\mathrm{T}}, \quad \textit{i.e.} \quad \mathbf{l}_i^{\mathrm{T}} B = \lambda_i \mathbf{l}_i^{\mathrm{T}} A.$$

Left multiplication of (B.1) by the ith eigenvector yields

$$\mathbf{l}_i^{\mathrm{T}}\left(A\frac{\partial \mathbf{u}}{\partial x}+B\frac{\partial \mathbf{u}}{\partial y}\right)=\mathbf{l}_i^{\mathrm{T}}\mathbf{c}$$
$$\mathbf{l}_i^{\mathrm{T}}A\left(\frac{\partial \mathbf{u}}{\partial x}+\lambda_i\frac{\partial \mathbf{u}}{\partial y}\right)=\mathbf{l}_i^{\mathrm{T}}\mathbf{c}.$$

Now consider differentiation alng a characteristic curve $\mathcal{C}(\tau)$. By the chain rule $\dot{\mathbf{u}}=\dot{x}_i\mathbf{u}_x+\dot{y}_i\mathbf{u}_y=\dot{x}_i(\mathbf{u}_x+\lambda_i\mathbf{u}_y)$, where a dot signifies differentiation w.r.t. τ, thus

$$\mathbf{l}_i^{\mathrm{T}}A\left(\frac{\partial \mathbf{u}}{\partial x}+\frac{\mathrm{d}y}{\mathrm{d}x}\frac{\partial \mathbf{u}}{\partial y}\right)=\mathbf{l}_i^{\mathrm{T}}\mathbf{c}\dot{x}$$
$$\mathbf{l}_i^{\mathrm{T}}A\dot{\mathbf{u}}=\mathbf{l}_i^{\mathrm{T}}\mathbf{c}\dot{x}.$$

What the above manipulation tells us is the following: if it is possible to arrange the PDE to the form, along the curve $\mathcal{C}(\tau)$

$$\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}\tau}=\frac{\mathrm{d}x}{\mathrm{d}\tau}\frac{\mathrm{d}\mathcal{R}}{\mathrm{d}x}=0,$$

for some function $\mathcal{R}$, then we say that $\mathcal{R}$ is a *Riemann invariant*[1] for this curve $\mathcal{C}(\tau)$. Therefore, if

$$\mathbf{l}_i^{\mathrm{T}}A\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}\tau}=\mathbf{l}_i^{\mathrm{T}}\mathbf{c} \tag{B.2}$$

can be integrated along such a curve $\mathcal{C}(\tau)$, this is equivalent to finding such an $\mathcal{R}$.

B.1.1 Linearisation

Linearisation is a standard technique used to approximate a system in a small neighbourhood, so that more complicated parts can be ignored. In this section we consider a wave of very small amplitude. Then the equations yield the well-known wave equation, which has solutions that are valid for small disturbances in the surface of the water.

We assume that there is no current in the water, but that there are waves of small amplitude (size h_0, about 5% of the depth of the water), so that

$$u(x,y,t)=U+u_0(x,y,t) \qquad \text{where } U=0,\ u_0=\mathcal{O}(\varepsilon) \text{ for } \varepsilon\ll 1,$$
$$v(x,y,t)=V+v_0(x,y,t) \qquad \text{where } V=0,\ v_0=\mathcal{O}(\varepsilon) \text{ for } \varepsilon\ll 1,$$
$$h(x,y,t)=H+h_0(x,y,t) \qquad \text{where } H=-b,\ h_0=\eta=\mathcal{O}(\varepsilon).$$

1. Equivalently, if $\mathrm{d}\mathcal{R}/\mathrm{d}x=c$ for some constant c, we may write $\mathrm{d}(\mathcal{R}-cx)/\mathrm{d}x=0$ then $\mathcal{R}-cx$ is a Riemann invariant.

The shallow water equations in vector calculus notation, with $\mathbf{u} = (u, v)$, are

$$h_t + \mathbf{u} \cdot \nabla h + h \nabla \cdot \mathbf{u} = 0, \tag{B.3a}$$

$$\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u} + g \nabla h = -g \nabla b. \tag{B.3b}$$

Substitute the linearised form in to (B.3), noting that $H_t = 0$, and neglecting the nonlinear advection terms $\mathbf{u}_0 \cdot \nabla(\cdot)$, to obtain

$$h_{0t} + (H + h_0) \nabla \cdot \mathbf{u}_0 = 0, \tag{B.4a}$$

$$\mathbf{u}_{0t} + g \nabla (H + h_0) = g \nabla H. \tag{B.4b}$$

Differentiate (B.4a) with respect to t, to obtain $h_{0tt} = -h_{0t} \nabla \cdot \mathbf{u}_0 - (H + h_0) \nabla \cdot \mathbf{u}_{0t}$. Then use the fact that $\mathbf{u}_{0t} = -g \nabla h_0$ to obtain $h_{0tt} = -h_{0t} \nabla \cdot \mathbf{u}_0 + g(H + h_0) \nabla^2 h_0$. Neglecting quadratic terms $h_{0t} \nabla \cdot \mathbf{u}_0$ and $h_0 \nabla^2 h_0$, we arrive at the wave equation

$$h_{0tt} = gH \nabla^2 h_0.$$

Suppose the initial data for the 1D problem is given as $u(x, 0) = \varphi(x)$, $u_t(x, 0) = \psi(x)$, then we obtain d'Alembert's formula for the one dimensional wave equation, with $c = \sqrt{gH}$,

$$u(x, t) = \frac{1}{2}[\varphi(x - ct) + \varphi(x + ct)] + \frac{1}{2c} \int_{x-ct}^{x+ct} \psi(\xi) \, \mathrm{d}\xi.$$

Hence we have shown that, for a wave of small amplitude with respect to the depth of the water, the wave follows the model of the well-known wave equation.

APPENDIX C

Asymptotic analysis

C.1 Green's functions for a circular domain

We use the method of images to construct a Green's function for a circular domain. The domain D is defined by $D = \{r \leqslant r_0 : 0 \leqslant r \leqslant r_0\}$, with ∂D_2, the inlet/outlet, at $(r_0 \cos\theta,\, r_0 \sin\theta)$ such that $\theta \in (-\beta,\, \beta)$ for some small value of β, and ∂D_1 the rest of the boundary. The Green's function G must satisfy

i) $\nabla^2 G(\mathbf{x}, \xi) = -\delta(\mathbf{x} - \xi)$ for $(r, \theta) \in D$,

ii) $\partial G/\partial n = 0$ for $(r, \theta) \in \partial D_1 \cup \partial D_2$,

i.e. it has Neumann boundary conditions.

Let $\mathbf{x} = (r, \theta) \in D$, $\xi = (\rho, \hat{\theta}) \in D$ be points inside D, and $\xi^* = (\rho^*, \hat{\theta}^*) \notin D$ be such that $\xi^* = \xi/|\xi^*|$ so that it is outside the domain. Then define distances R^2 and R^{*2} as

$$R^2 = |\mathbf{x} - \xi|^2 = r^2 + \rho^2 - 2r\rho\cos(\theta - \hat{\theta}),$$
$$R^{*2} = |\mathbf{x} - \xi^*|^2 = r^2 + \frac{1}{\rho^2} - 2\frac{r}{\rho}\cos(\theta - \hat{\theta}).$$

The Green's function that satisfies $\nabla^2 G = -\delta$ in D and $\partial G/\partial n = 0$ in $\partial D_1 \cup \partial D_2$ is

$$\frac{\partial G(\mathbf{x}, \xi)}{\partial n} = -\frac{1}{2\pi}\ln\left(\frac{R^*}{R}\right) = -\frac{1}{4\pi}\ln\left[\frac{\left(r^2 + \frac{1}{\rho^2} - 2r\frac{1}{\rho}\cos(\theta - \hat{\theta})\right)}{r^2 + \rho^2 - 2r\rho\cos(\theta - \hat{\theta})}\right]. \tag{C.1}$$

Thus for the flux of the shallow water system with source terms we have

$$\mathbf{p} = (p,\, q) = \varepsilon \nabla \psi_1 + \varepsilon^2 \nabla \psi_2 + \mathcal{O}(\varepsilon^4),$$

where ∇ is the usual grad in polar coordinates $\nabla \psi = \frac{\partial \psi}{\partial r}\mathbf{e}_r + \frac{\partial \psi}{\partial \theta}\mathbf{e}_\theta$, and the flux functions are

$$\psi_1 = \int_D G h_{1t}\,\mathrm{d}V + \int_{\partial D_2} G k_1\,\mathrm{d}S = \sin(t)\int_{r=0}^{1}\int_{\theta=0}^{2\pi} G\, r \mathrm{d}r\mathrm{d}\theta + \int_{3\pi/2-\beta}^{3\pi/2+\beta} G k_1\,\mathrm{d}\theta$$
$$\psi_2 = \int_{\partial D_2} G k_2\,\mathrm{d}S = \int_{3\pi/2-\beta}^{3\pi/2+\beta} G k_2\,\mathrm{d}\theta$$

with

$$k_1 = -\mathbf{u}_1 b, \quad k_2 = \mathbf{u}_1 \phi_1.$$

For the surface elevation for the shallow water equations with source terms we have

$$h = -b + \varepsilon \phi_1 + \mathcal{O}(\varepsilon^4)$$

where b represents the bottom slope and ϕ_1 represents the forcing of the tide.

C.2 Stratified shallow water

The water in a bay may become stratified through temperature or density. Variations in temperature are likely to be negligible in shallow water, but may be seen in a deep lake, sea or in a dam.

Stratification of the fluid via density differences in a bay of shallow water may occur in a number of natural ways – through melting icesheets in spring, through freshwater rivers entering saltier oceans, through tidal exchanges between seas. This mechanism is likely to result in a stably stratified fluid with no vertical mixing. An unstably stratified fluid can occur when there is instability even without flow, for example in bays with a high evaporation rate such as the Mediterranean Sea.

The stratified shallow water problem is mathematically interesting from the point of view that the system no longer has easily-obtainable eigenvalues. For this reason, Riemann invariants cannot be found as easily as in the single layer case, and so analytical solutions are more elusive

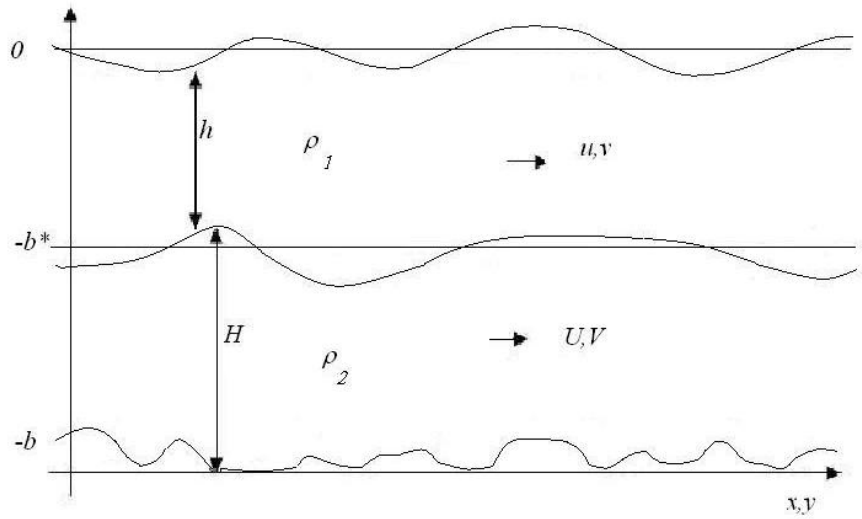


Figure C.1: Schematic of geometry of a bay of stratified water.

even in simple cases *e.g.* when there are no shocks. Baines (1995) gives a comprehensive mathematical treatment including waves in two-layer fluids and flow over irregular (2D and 3D) topography.

In this section we consider a domain where fluid is horizontally stratified due to differences in density.

C.2.1 Analysis

Let the geometry of the bay be as shown in Figure C.1. The depth of the upper layer is denoted h, with velocities u, v in the x- and y-directions respectively (the flux vector $\mathbf{p} = (hu, hv)$), and the density of the upper layer is denoted ρ_1. In the upper layer, let the depth be denoted by H, the x- and y-velocities by U and V (the flux vector by $\mathbf{P} = (HU, HV)$), and the density by ρ_2.

The governing equations are

$$h_t + \nabla \cdot \mathbf{p} = 0, \tag{C.2a}$$

$$H_t + \nabla \cdot \mathbf{P} = 0, \tag{C.2b}$$

$$\mathbf{p}_t + \nabla \cdot \left(\frac{\mathbf{p}^2}{h}\right) + \left(\frac{\partial}{\partial y}, \frac{\partial}{\partial x}\right)\left(\frac{pq}{h}\right) + gh\nabla(h+H) = -gh\nabla b, \tag{C.2c}$$

$$\mathbf{P}_t + \nabla \cdot \left(\frac{\mathbf{P}^2}{H}\right) + \left(\frac{\partial}{\partial y}, \frac{\partial}{\partial x}\right)\left(\frac{PQ}{H}\right) + gH\nabla(\gamma h+H) = -gH\nabla b, \tag{C.2d}$$

where $\gamma = \rho_1/\rho_2$. Making the assumption that $\gamma \approx 1$ is known as the Boussinesq approximation.

Eigenvalues in two dimensions

The eigenvalues of this system are not easily determined analytically. However, it is interesting to see how far it is possible to take the analysis, to see how far similarities go between the stratified and unstratified systems. To determine the eigenvalues of (C.2), write the system in matrix form as follows

$$\mathbf{q}_t + A\mathbf{q}_x + B\mathbf{q}_y = \mathbf{s}, \tag{C.3}$$

where

$$\mathbf{q} = \begin{bmatrix} h \\ H \\ hu \\ HU \\ hv \\ HV \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ gh-u^2 & gh & 2u & 0 & 0 & 0 \\ \gamma gH & gH-U^2 & 0 & 2U & 0 & 0 \\ -uv & 0 & v & 0 & u & 0 \\ 0 & -UV & 0 & V & 0 & U \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -uv & 0 & v & 0 & u & 0 \\ 0 & -UV & 0 & V & 0 & U \\ gh-v^2 & gh & 0 & 0 & 2v & 0 \\ \gamma gH & gH-V^2 & 0 & 0 & 0 & 2V \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} 0 \\ 0 \\ ghb_x \\ gHb_x \\ ghb_y \\ gHb_y \end{bmatrix}.$$

Choose a normal vector $\mathbf{n} = (n_x, n_y)$ such that $n_x^2 + n_y^2 = 1$, and calculate the eigenvectors λ_i for $i = 1, 2, \ldots, 6$ of $n_xA + n_yB$ such that $|(n_xA + n_yB) - \lambda_i| = 0$

$$(\lambda_i - (n_xu + n_yv))(\lambda_i - (n_xU + n_yV))\sum_{k=0}^{4} a_k\lambda_i^k = 0 \tag{C.4}$$

where the a_k are

$$
\begin{aligned}
a_4 &= 1 \\
a_3 &= -4\,(n_x\bar{u} + n_y\bar{v}) \\
a_2 &= 4(n_x\bar{u} + n_y\bar{v})^2 + 2(n_xu + n_yv)(n_xU + n_yV) - g(h + H) \\
a_1 &= -4\,(n_x\bar{u} + n_y\bar{v})\,(n_xu + n_yv)(n_xU + n_yV) + 2g(h + H)(n_x\hat{u} + n_y\hat{v}) \\
a_0 &= (n_xu + n_yv)^2(n_xU + n_yV)^2 - gh(n_xU + n_yV) - gH(n_xu + n_yv) - g^2(1 - \gamma)hH \\
&= -(1 - \gamma)g^2hH(1 - \mathrm{G}^2),
\end{aligned}
$$

and $\bar{u} = \frac{1}{2}(u + U)$, $\bar{v} = \frac{1}{2}(v + V)$, $\hat{u} = \frac{1}{(h+H)}(hU + Hu)$, $\hat{v} = \frac{1}{(h+H)}(hV + Hv)$, and where the composite Froude number (Lawrence (1981), and Armi (1986)) is

$$
\mathrm{G}^2 = \frac{(n_xu + n_yv)^2}{g(1 - \gamma)h} + \frac{(n_xU + n_yV)^2}{g(1 - \gamma)H} + \frac{(n_xu + n_yv)^2(n_xU + n_yV)^2}{g^2(1 - \gamma)hH}.
$$

It can be seen how these coefficients make (C.4) reduce to the eigenvalues for the unstratified system when variables in the top and bottom layers are equal.

Two of the roots of (C.4) are $\lambda_i = \mathbf{n}\cdot\mathbf{u}$, $\mathbf{n}\cdot\mathbf{U}$, and the remaining four are found by solving the quartic polynomial $\sum_{k=0}^{4} a_k\lambda_i^k = 0$. Since this quartic has in general no analytic solutions, we look at some methods for approximating the eigenvalues.

Operator splitting Salmon (2002) has treated the 1D stratified shallow water equations by 'splitting' the flux matrix to be a sum of one containing advective terms, and the other corresponding to a linearisation about a steady state. We extend the idea to two spatial dimensions by writing (C.3) as

$$
\mathbf{q}_t + (A_1 + A_2)\mathbf{q}_x + (B_1 + B_2)\mathbf{q}_y = \mathbf{s},
$$

with

$$n_xA_1 + n_yB_1 = \begin{bmatrix} \mathbf{n}\cdot u & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{n}\cdot U & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{n}\cdot u & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{n}\cdot U & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{n}\cdot u & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{n}\cdot U \end{bmatrix},$$

$$n_xA_2 + n_yB_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & n_xg & n_xg \\ 0 & 0 & 0 & 0 & n_x\gamma g & n_xg \\ 0 & 0 & 0 & 0 & n_yg & n_yg \\ 0 & 0 & 0 & 0 & n_y\gamma g & n_yg \\ n_xh & 0 & n_yh & 0 & 0 & 0 \\ 0 & n_xH & 0 & n_yH & 0 & 0 \end{bmatrix}$$

The eigenvalues of $n_xA_1 + n_yB_1$ are easily seen to be $\mathbf{n}\cdot\mathbf{u}$ and $\mathbf{n}\cdot\mathbf{U}$, which are real, but not necessarily distinct. The eigenvalues of $n_xA_2 + n_yB_2$ are the λ_i that satisfy $|(n_xA_2 + n_yB_2) - \lambda_i| = 0$, so that

$$\lambda_i^2 = 0$$
$$\lambda_i^2 = \frac{1}{2}g\left(h + H(n_x^2 + \gamma n_y^2)\right) \pm \frac{1}{2}g\sqrt{\left(h + H(n_x^2 + \gamma n_y^2)\right)^2 - 4hH(1-\gamma)(n_x^2 - n_y^2)},$$

which agrees with the result for the one-dimensional case from Salmon (2002), where

$$\lambda^2 = \tfrac{1}{2}g(h+H) \pm \tfrac{1}{2}g\sqrt{(h+H)^2 - 4hH(1-\gamma)}.$$

Assume $u = U$, $v = V$ Under the assumption that the velocities in the upper and lower layers were equal, Lawrence (1981) gave exact solutions to the one-dimensional system. Using that idea, putting $u = U$ and $v = V$ in (C.4), we obtain the eigenvalues for the corresponding 2D system

$$\lambda_E^{\pm} = \tfrac{1}{2}\bar{\mathbf{u}} \pm \sqrt{\tfrac{1}{2}g(h+H)\left(1 + \sqrt{1 - \frac{4(1-\gamma)hH}{(h+H)^2}}\right)}$$
$$\lambda_I^{\pm} = \tfrac{1}{2}\bar{\mathbf{u}} \pm \sqrt{\tfrac{1}{2}g(h+H)\left(1 - \sqrt{1 - \frac{4(1-\gamma)hH}{(h+H)^2}}\right)},$$

where $\bar{\mathbf{u}} = \left(\frac{1}{2}(u+U), \frac{1}{2}(v+V)\right)$. The two eigenvalues $\lambda_E^{\pm}$ correspond to external (free-surface) wave motions, and the $\lambda_I^{\pm}$ correspond to internal (interfacial) wave motions (Lawrence (1981)).

Riemann2D A numerical method for calculating the eigenvalues is needed if *Riemann2D* is to be extended to include a stratified model. Muñoz-Ruiz *et al.* (2000) have used a Newton-Raphson method to numerically solve for the eigenvalues, which could be programmed into a new extension for *Riemann2D* . A library exists called JAMA (Java Matrix package), developed by Cleve Moler [1] and available to download freely. This package is imported into the required classes in *Riemann2D* and all its functionality – such as finding eigenvalues, eigenvectors, decomposing a matrix etc – are available to the programmer. This is in fact a good demonstration of the advantages of OOP – it can be very easy to develop packages in Java that fit in with existing code to extend its usability.

C.2.2 Asymptotic analysis for stratified flow

Nondimensionalisation

We work with the assumption that the order of magnitude of the velocities are the same in each layer, and each layer is the same order of magnitude in depth. In the same way as for the unstratified model, *i.e.* using the nondimensional variables

$$h, H = D\tilde{h}, \tilde{H}, \quad u, U = U_*\tilde{u}, \tilde{U}, \quad v, V = U_*\tilde{v}, \tilde{V}, \quad t = T\tilde{t},$$
$$x = U_*T\tilde{x}, \quad y = U_*T\tilde{y}, \quad b = D\tilde{b}, \quad \text{(C.5)}$$

we find a Froude number $\mathrm{Fr}^2 = gD/U_*$, where U_* is a typical size of the velocities u, v, U and V. Then we exploit the smallness of the Froude number to allow asymptotic analysis to be carried out.

The nondimensionalised, two-dimensional shallow water equations with no source terms are

$$h_t + \nabla \cdot \mathbf{p} = 0 \quad \text{(C.6a)}$$

$$H_t + \nabla \cdot \mathbf{P} = 0 \quad \text{(C.6b)}$$

$$\varepsilon^4 \left(\mathbf{p}_t + \nabla \cdot \left(\frac{\mathbf{p}^2}{h} \right) + \left(\frac{\partial}{\partial y}, \frac{\partial}{\partial x} \right) \left(\frac{pq}{h} \right) \right) + h\nabla(h + H) = -h\nabla b \quad \text{(C.6c)}$$

$$\varepsilon^4 \left(\mathbf{P}_t + \nabla \cdot \left(\frac{\mathbf{P}^2}{H} \right) + \left(\frac{\partial}{\partial y}, \frac{\partial}{\partial x} \right) \left(\frac{PQ}{H} \right) \right) + H\nabla(\gamma h + H) = -H\nabla b. \quad \text{(C.6d)}$$

1. `http://math.nist.gov/javanumerics/jama/`

Boundary conditions

Let the domain D have boundary $\partial D = \partial D_1 \cup \partial D_2 \cup \partial D_3$ where ∂D_1 represents the wall of the bay, ∂D_2 is the ocean boundary, and ∂D_3 is the river boundary. We assume no flow through the wall of the bay so that

$$\mathbf{p} \cdot \mathbf{n} = 0 \quad \text{and} \quad \mathbf{P} \cdot \mathbf{n} = 0 \quad \text{for} \quad x, y \in \partial D_1. \tag{C.7}$$

At the ocean boundary, seawater enters the bay, and since it is more dense than the freshwater, it sits underneath. Assume the seawater enters in a tidal fashion, and that the tide alters the water-level by an amount $\mathcal{O}(\varepsilon)$. Assume that flux is specified across this boundary the depth of the lower layer of water is

$$H(x, y, t) = -b(x, y) + b_* + \varepsilon \Phi_1(t), \tag{C.8a}$$

$$\mathbf{P}(x, y, t) \cdot \mathbf{n} = K(x, y, t) = K_0(x, y, t) + \varepsilon K_1(x, y, t) + \dots \tag{C.8b}$$

$$\mathbf{p}(x, y, t) \cdot \mathbf{n} = 0 \quad \text{for} \quad x, y, \in \partial D_2 \tag{C.8c}$$

where $\Phi_1(t)$ is some function that represents the periodic nature of the tide, and $K(x, y, t) = K_0 + \varepsilon K_1 + \dots$ is a flux function in which we have $K_0 = -\mathbf{U}_0(b - b_*)$, $K_1 = \mathbf{U}_1(b - b_*) + \mathbf{U}_0\Phi_1$, $K_2 = \mathbf{U}_1\Phi_1$, and $K_3 = 0$, according to (3.23).

At the river mouth, the amount of water being contributed to the bay is assumed to be smaller than the amount brought by the ocean.

$$h(x, y, t) = -b_* + \varepsilon^2 \phi_2 \tag{C.9a}$$

$$\mathbf{p}(x, y, t) \cdot \mathbf{n} = k(x, y, t) = k_0 + \varepsilon k_1 + \dots \tag{C.9b}$$

$$\mathbf{P}(x, y, t) \cdot \mathbf{n} = 0 \quad \text{for} \quad x, y, \in \partial D_3, \tag{C.9c}$$

where ϕ_2 is a function of time that represents the inflow/outflow of water and that is not necessarily periodic, and $k(x, y, t) = k_0 + \varepsilon k_1 + \dots$ is a flux function with $k_0 = -\mathbf{u}_0 b_*$, $k_1 = -\mathbf{u}_1 b_*$, $k_2 = \mathbf{u}_0\phi_2 - \mathbf{u}_2 b_*$, and $k_3 = 0$.

Let $h = h_0 + \varepsilon h_1 + \dots$, $u = u_0 + \varepsilon u_1 + \dots$ and expand system (C.6) in terms of ε. Group terms in orders of ε and consider each in turn.

Leading order

The momentum equations (C.6c) and (C.6d) are

$$h_0\nabla(h_0 + H_0 + b) = 0, \quad \text{and} \quad H_0\nabla(\gamma h_0 + H_0 + b) = 0.$$

For $h_0 \neq 0$, $H_0 \neq 0$, we have

$$h_0 + H_0 + b = \phi_0(t), \quad \text{and} \quad \gamma h_0 + H_0 + b = \Phi_0(t).$$

The boundary conditions on h (C.9a) and H (C.8a) say that at leading order $h_0 = -b_*$ and $H_0 = -b + b_*$ on both ∂D_2 and ∂D_3. This implies $\phi_0 = 0$ and $\Phi_0 = (1-\gamma)b_*$, so we must have $h_0 = -b_*$, $H_0 = -b + b_*$ everywhere.

From the equations for mass (C.6a) and (C.6b) we have

$$\nabla \cdot \mathbf{p_0} = -h_{0t}, \quad \text{and} \quad \nabla \cdot \mathbf{P_0} = -H_{0t},$$

then define $\mathbf{p}_0 = \nabla A_0$, $\mathbf{P}_0 = \nabla B_0$, so

$$\nabla^2 A_0 = 0, \quad \text{and} \quad \nabla^2 B_0 = 0,$$

with boundary conditions

$$\frac{\partial A_0}{\partial n} = 0 \text{ on } \partial D_1, \partial D_2, \quad \frac{\partial A_0}{\partial n} = k_0 \text{ on } \partial D_3, \quad \text{and}$$
$$\frac{\partial B_0}{\partial n} = 0 \text{ on } \partial D_1, \partial D_3, \quad \frac{\partial B_0}{\partial n} = K_0 \text{ on } \partial D_2.$$

But by the Hopf maximum principle, as explained in Chapter 3, we have that flux is zero in upper and lower layers, *i.e.*

$$\mathbf{p}_0 = (p_0, q_0) = \mathbf{0}, \quad \text{and} \quad \mathbf{P}_0 = (P_0, Q_0) = \mathbf{0}.$$

Order ε

Since $\nabla(h_0 + H_0 + b) = 0 = \nabla(\gamma h_0 + H_0 + b)$, the momentum equations (C.6c) and (C.6d) at order ε are

$$h_0\nabla(h_1 + H_1) = 0, \quad \text{and} \quad H_0\nabla(\gamma h_1 + H_1) = 0.$$

For $h_0 \neq 0$, $H_0 \neq 0$,

$$h_1 + H_1 = \phi_1(t), \quad \text{and} \quad \gamma h_1 + H_1 = \Phi_1(t)$$

for some unknown functions ϕ_1 and Φ_1. At the ocean boundary ∂D_2, $h_1 = 0$ and $H_1 = \Phi_1(t)$, where $\Phi_1(t)$ is a known function. Hence $h_1 = 0$ and $H_1 = \Phi_1(t)$ everywhere.

From the equations for conservation of mass (C.6a) and (C.6b),

$$\nabla \cdot \mathbf{p_1} = -h_{1t}, \quad \text{and} \quad \nabla \cdot \mathbf{P_1} = -H_{1t}$$
$$\nabla^2 A_1 = 0, \quad \text{and} \quad \nabla^2 B_1 = -\frac{\partial \Phi_1}{\partial t},$$

where $\mathbf{p}_1 = \nabla A_1$, $\mathbf{P}_1 = \nabla B_1$, with boundary conditions

$$\frac{\partial A_1}{\partial n} = 0 \text{ on } \partial D_1, \partial D_2, \quad \frac{\partial A_1}{\partial n} = k_1 \text{ on } \partial D_3 \quad \text{and}$$
$$\frac{\partial B_1}{\partial n} = 0 \text{ on } \partial D_1, \partial D_3, \quad \frac{\partial B_1}{\partial n} = K_1 \text{ on } \partial D_2.$$

These can be solved by looking for a Green's function, G, as before that satisfies

i) $\nabla^2 G = -\delta(\mathbf{x}, \xi)$ on D

ii) $\partial G/\partial n = 0$ on ∂D.

Using Green's second identity,

$$A_1(\mathbf{x}) = \int_{\partial D_3} G(\mathbf{x}, \xi) k_1 \, \mathrm{d}S, \quad \text{and} \quad B_1(\mathbf{x}) = \int_D G(\mathbf{x}, \xi) H_{1t} \, \mathrm{d}V + \int_{\partial D_2} G(\mathbf{x}, \xi) K_1 \, \mathrm{d}S.$$

Order ε^2

Since $\nabla(h_0 + H_0 + b) = 0 = \nabla(\gamma h_0 + H_0 + b)$ and $h_0 = -b_*$, $h_1 = 0$ we have the momentum equations (C.6c) and (C.6d) at $\mathcal{O}(\varepsilon^2)$

$$h_0 \nabla(h_2 + H_2) = 0, \quad \text{and} \quad H_0 \nabla(\gamma h_2 + H_2) = 0.$$

Then for $h_0 \neq 0$, $H_0 \neq 0$, we obtain

$$h_2 + H_2 = \phi_2(t), \quad \text{and} \quad \gamma h_2 + H_2 = \Phi_2(t).$$

At the river inlet ∂D_3, the boundary conditions (C.9a) are $h_2 = \phi_2(t)$, $H_2 = 0$. Thus $h_2 = \phi_2 = \Phi_2/\gamma$. Now from conservation of mass (C.6a) and (C.6b)

$$\nabla^2 A_2 = -h_{2t}, \quad \text{and} \quad \nabla^2 B_2 = 0,$$

where $\nabla A_2 = (p_2, q_2)$, $\nabla B_2 = (P_2, Q_2)$. Together with boundary conditions $\partial A_2/\partial n = 0$ on $\partial D_1 \cup \partial D_2$ and $\partial A_2/\partial n = k_2$ on ∂D_3 and $\partial B_2/\partial n = 0$ on $\partial D_1 \cup \partial D_2 \cup \partial D_3$, we use Green's identities to obtain

$$A_2 = \int_D G h_{2t} \,\mathrm{d}V + \int_{\partial D_3} G k_2 \,\mathrm{d}S, \quad \text{and} \quad B_2 = \int_{\partial D_2} K_2 G \,\mathrm{d}S.$$

Order ε^3

The equations of momentum (C.6c) and (C.6d) at $\mathcal{O}(\varepsilon^3)$ reduce to

$$h_0 \nabla(h_3 + H_3) = 0, \quad \text{and} \quad H_0 \nabla(\gamma h_3 + H_3) = 0.$$

For $h_0 = 0 = H_0$, we can write

$$\nabla(h_3 + H_3) = 0, \quad \text{and} \quad \nabla(\gamma h_3 + H_3) = 0$$
$$h_3 + H_3 = \phi_3 \quad \text{and} \quad \gamma h_3 + H_3 = \Phi_3.$$

The boundary conditions on h_3, H_3 are $h_3 = 0 = H_3$ on ∂D, *i.e.* on all of the boundary. Therefore $h_3 = 0$, $H_3 = 0$ everywhere.

Now from the equations for conservation of mass (C.6a) and (C.6b) we find that

$$\nabla^2 A_3 = 0, \quad \text{and} \quad \nabla^2 B_3 = 0.$$

Solve these equations by looking for a Green's function G_3 and using boundary conditions $\partial A_3/\partial n = 0$ on ∂D and $\partial B_3/\partial n = 0$ on ∂D, so we obtain

$$A_3 = \int_{\partial D_3} k_3 G \,\mathrm{d}V, \quad \text{and} \quad B_3 = \int_{\partial D_2} K_3 G \,\mathrm{d}V.$$

But since $k_3 = 0 = K_3$, we have

$$\mathbf{p}_3 = \mathbf{0}, \quad \text{and} \quad \mathbf{P}_3 = \mathbf{0}.$$

Solution up to order ε^3

In total, the solution up to order ε^3 is the following

$$
\begin{aligned}
h &= -b_* + \varepsilon^2 \phi_2(t) + \mathcal{O}(\varepsilon^4) \\
H &= -b + b_* + \varepsilon \phi_1(t) + \mathcal{O}(\varepsilon^4) \\
\mathbf{p} &= \varepsilon \nabla \int_{\partial D_3} G k_1 \,\mathrm{d}S + \varepsilon^2 \nabla \left(\int_D G h_{2t} \,\mathrm{d}V + \int_{\partial D_3} G k_2 \,\mathrm{d}S \right) + \mathcal{O}(\varepsilon^4) \\
\mathbf{P} &= \varepsilon \nabla \int_D G H_{1t} \,\mathrm{d}V + \varepsilon^2 \nabla \int_{\partial D_2} G K_2 \,\mathrm{d}S + \mathcal{O}(\varepsilon^4).
\end{aligned}
$$

It can be seen how similar these solutions are to the single-layer solutions.

APPENDIX D

Eclipse

Eclipse is an IDE (integrated development system) that has versions available for Java, C++ and C. It is free to download from the internet from `www.eclipse.org`. The alternative to using an IDE is using a text editor like Notepad or Gedit and saving the file with a .java extension and running the code at the command line. Eclipse has many features that makes programming much easier than this option.

Auto-complete allows you to fill in the word you started typing with valid options. This reduces the risk of writing code that will not compile.

Error highlighting underlines any code that will not compile with a red line and a red cross in the margin, so that the errors can be easily located and resolved before compiling the code.

Field linking highlights all instances of a chosen field within the open class, making it easy to locate a variable that might be dealt with differently in another part of the class and so reducing conflicts.

Debug mode allows breakpoints to be set and goes through the code sequentially. This allows you to see what parts of the code are not being reached. When the mouse is held over a variable during debug, a box is displayed that contains the information of that variable at that time.

History view means you can compare the current version of the class with another version that is automatically saved by Eclipse. The differences are highlighted and there is a restore button that automatically takes the current version back to the programmer's selected version. This saves time and is more accurate than trying to remember what was written and compiled well a few days ago and restore it by memory.

Team repository allows members of the team to submit new improved code and check out new versions of the code that other team members have submitted. It is a way of communicating, which is essential when a group of two or more people are working on a relatively complex piece of code.

Eclipse is user-friendly and has many advantages. There are many help pages on the internet and Eclipse itself has some useful 'getting started' tutorials. There are other IDEs in use such as JBuilder and NetBeans, more information on which can be found on the world wide web.

References

Abdennadher, J. and Boukthir, M. (2006). Numerical simulation of the barotropic tides in the Tunisian Shelf and the Strait of Sicily. *J. of Marine Systems*, **63**, 162–182. 7.3.1

Acheson, D. (1990). *Elementary fluid dynamics*. O.U.P. 5.3.1

Alcrudo, F. and Benkhaldoun, F. (2001). Exact solutions to the Riemann problem of the shallow water equations with a bottom step. *Computers and Fluids*, **30**, 643–671. 3.1.2, 3.1.3

Anastasiou, K. and Chan, C. (1997). Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes. *Int. J. for Num. Meth. in Fluids*, **24**, 1225–1245. 2.2.6

Anderson, B. (2002). On fluvial resistance. In R. Mein, ed., *Catchworld: newsletter of the cooperative research centre for catchment hydrology*, 16–18, Cooperative research centre for catchment hydrology. 6.2

Armi, L. (1986). The hydraulics of two flowing layers with different densities. *J. Fluid Mechanics*, **163**, 27–58. C.2.1

Baines, P. (1974). On the drag coefficient over shallow water. *Boundary-layer meteorology*, **6**, 299–303. 2.1.1, 3.2.4

Baines, P. (1995). *Topographic effects in stratified flows*. C.U.P. C.2

Batten, P., Lambert, C. and Causon, D. (1996). Positively conservative high-resolution convective schemes for unstructured elements. *Int. J. for Num. Meth. in Eng.*, **39**, 1821–1838. 2.2.3, 4.1.4

Becker, M. and Jiang, Z. (2007). Flux-based contaminant transport in a GIS environment. *J. of Hydr*, **343**, 203–210. 2.5.1

Benkhaldoun, F., Elmahi, I. and Seaïd, M. (2007). Well-balanced finite volume schemes for pollutant transport by shallow water equations on unstructured meshes. *J. of Comp. Phys*, **226**, 180–203. 2.3.1, 2.3.1, 4.3.4, 6.1.3

Bermúdez, A. and Vázquez, M. (1994). Upwind methods for hyperbolic conservation laws with source terms. *Computers Fluids*, **23**, 1049–1071. 2.2.3, 3.1.3, 4.1.3

Bermúdez, A., Dervieux, A., Desideri, J. and Vázquez, M. (1998). Upwind schemes for the two-dimensional shallow water equations with variable depth using unstructured meshes. *Comput. Methods Appl. Mech. Engrg.*, 49–72. 2.2.3, 2.2.6, 2.3, 2.3

Black, K., Hatton, D. and Rosenberg, M. (1993). Locally and externally-driven dynamics of a large semi-enclosed bay in southern Australia. *J. Coastal Res.*, **2**, 509–538. 6.2, 6.2.1

Bohr, T., Dimon, P. and Putkaradze, V. (1993). Shallow-water approach to the circular hydraulic jump. *J. Fluid Mech.*, **254**, 635–648. 2.1.2

Bolton, G. (1981). *Soils and spoilers*. George Allen & Unwin. 2.4.1

Borel, A., Henkin, G. and Lax, P. (2000). Jean leray (1906-1998). *Notices of the AMS*, **47**, 350–359. 3.1.4

Boris, J. and Book, D. (1973). Flux-corrected transport: I SHASTA a fluid transport algorithm that works. *J. of Comp. Phys*, **11**, 38–69. 2.2.3, 2, 4.1.4

Borthwick, A., Cruz Léon, S. and Józsa, J. (2001a). Adaptive quadtree model of shallow-flow hydrodynamics. *J. of Hyr. Res.*. 2.1.1, 2.1.2, 2.2.6, 4.1.5, 5.4, 5.11, 5.5, 8.1.3

Borthwick, A., Fujihara, M. and Rogers, B. (2001b). *In Godunov Methods: Theory and Applications*, chap. Godunov-type solution of shallow water equations on curvilinear and quadtree grids, 141–148. Kluwer. 2.1.2, 2.2.6, 4.1.5, 5.4, 5.11, 5.5, 8.1.3

Boyd, J. (2000). *Chebyshev and Fourier spectral methods*. Dover, 2nd edn., www.math.ucl.ac.be/membres/magnus/num1a/boyd01.pdf. 2.2

Bradford, S. and Sanders, B. (2002). Finite-volume model for shallow-water flooding of arbitrary topography. *Journal of Hydraulic Engineering*, 289–298. 2.2.4, 2.3

Brufau, P. and Garcia-Navarro, P. (2000). Two-dimensional dam break flow simulation. *Int. J. Numer. Meth. Fluids*, **33**, 35–57. 2.2.6, 3.1.1, 4.1.2, 4.1.2

Butman, B. (1978). *On the dynamics of shallow water currents in Massachusetts Bay and on the New England continental shelf*. Ph.D. thesis, MIT and WHOI. 2.1.1, 3.2.4

Butman, B., Sherwood, C. and Dalyander, P. (2008). Northeast storms ranked by wind stress and wave-generated bottom stress observed in Masachusetts Bay, 1990-2006. *Continental Shelf Res.*, **28**, 1231–1245. 2.3.1, 2.3.1, 7.4.2, 7.5, 8.1.4

Cao, Z., Pender, G., Wallis, S. and Carling, P. (2004). Computational dam-break hydraulics over erodible sediment bed. *J. Hydr. Eng.*, 689–703. 8.2.7

Castro, M., García-Rodrígues, J., González-Vida, J., Macías, J., Parès, C. and Vázquez-Cendón, M. (2004). Numerical simulation of two-layer shallow water flows through channels with irregular geometry. *J. of Comp. Phys.*, **195**, 202–235. 7.3.1

Castro, M., García-Rodrígues, J., González-Vida, J. and Parès, C. (2006). A parallel 2D finite volume scheme for solving systems of balance laws with non-conservative products: Application to shallow flows. *Comput. Methods Appl. Mech. Eng.*, **195**, 2788–2815. 2.3

Castro, M., García-Redríguez, J., González-Vida, J., Macías, J. and Parès, C. (2007). Improved FVM for two-layer shallow-water models: Application to the Strait of Gibraltar. *Adv. in Num. Meth. for Env. Eng.*. 2.3

Chanson, H. (1998). Extreme reservoir sedimentation in Australia: a review. *Int. J. of Sediment Research*, **13**, 55–63. 2.4.2

Costanzo, S., ODonoghue, M., Dennison, W., Longeran, N. and Thomas, M. (2001). A new approach for detecting and mapping sewage impacts. *Marine Poll. Bull.*, **42**, 149–156. 7.4.2

Costanzo, S., Udy, J. and Jones, A. (2005). Using nitrogen stable isotope ratios of marcoalgae to determine the effectiveness of sewage upgrades: changes in the extent of sewage plumes over four years in Moreton Bay, Australia. *Marine Poll. Bull.*, **51**, 212–217. (document), 7.4.2, 7.9, 7.5

Courant, R., Freidrichs, K. and Lewy, H. (1928). On the partial difference equations of mathematical physics. *Reprinted in IBM Journal in 1967 from Mathematische Annalen*, **100**, 32–74. 2.2, 4.1.3

Dietel, H. (2005). *Java How To Program*. Pearson Education Inc., 6th edn. 2.5, 2.5.1

Dunn, R., Lemckert, C., Teasdale, P. and Welsh, D. (2007). Distribution of nutrients in surface and sub-surface sediments of Coombabah Lake, southern Moreton Bay (Australia). *Marine Pollution Bulletin*, **54**, 602–625. 2.4.1, 2.4.3, 7.1

Elshorbagy, A. and Ormsbee, L. (2006). Object-oriented modeling approach to surface water quality management. *Env. Modelling and Software*, **21**, 689–698. 2.5.1

Etling, D., Detering, H. and Theunert, F. (1985). On the simulation of wind-driven currents in shallow water. *Archives for meteorology , geophysics and bioclimatology*, **33**, 355–363. 2.1.1

Eyre, B., Hossain, S. and McKee, L. (1998). A suspended sediment budget for the modified subtropical Brisbane river estuary, Australia. *Estuarine, Coastal and Shelf Science*, **47**, 513–522. 7.3.2

Farmer, D. and Armi, L. (1988). The flow of Atlantic water through the Strait of Gibraltar. *Prog. Oceanog.*, **21**, 1–105. 2.3, 3.2.7

Feng, J., Cai, L. and Xie, W. (2006). CWENO-type central-upwind schemes for multidimensional Saint-Venant system of shallow water equations. *Applied Numerical Mathematics*, **56**, 1001–1017. 2.2.6, 2.3

Font, J.A. (2003). Numerical hydrodynamics in general relativity. *Living Reviews in Relativity*, **6**, cited on 1.04.2008. 4.1.3

FOWLER, A. (1997). *Mathematical models in the applied sciences*. C.U.P. 2.1.1, 3.2.1, 3.2.4

GELLER, S. (2003). Object-oriented modelling of an adaptive, quadtree-based finite volume method for the shallow water equations. www.irmb.tu-bs.de/UPLOADS/geller/OOMforSWE.pdf. 4.1.5

GLAISTER, P. (1988a). An approximate linearised Riemann solver for the Euler equations for real gases. *J. of Comp. Phys.*, **74**, 382–408. 2.2.4, 3.1.3, 4.1.1

GLAISTER, P. (1988b). An approximate linearised Riemann solver for the three-dimensional Euler equations for real gases using operator splitting. *J. of Comp. Phys.*, **77**, 361–383. 3.1.3

GLAISTER, P. (1990). Approximate Riemann solutions of the two-dimensional shallow water equations. *J. Eng. Math.*, **24**, 45–53. 2.2.2

GÓMEZ-VALDÉS, J., DELGADO, J. AND DWORAK, J. (2003). Overtides, compound tides, and tidal-residual current in Ensenada da la Paz lagoon, Baja California Sur, Mexico. *Geofisica Internacional*. 2.3

GREENBERG, D., SHORE, J., PAGE, F. AND DOWD, M. (2005). A finite element circulation model for embayments with drying intertidal areas and its application to the Quoddy region of the Bay of Fundy. *Ocean Modelling*, **10**, 211–231. 2.3, 7

HAFEZ, M. AND DIMANLIG, A. (1996). Some anomalies of the numerical solutions of the Euler equations. *Acta Mechanica*, **119**, 131–139. 5.3.1

HARTEN, A. (1983). High resolution schemes for hyperbolic conservation laws. *J. of Comp. Phys.*, **49**, 357–393. 2.2.3

HAYNES, D. (2001). *Pesticide and Heavymetal Concentrations in Great Barrier Reef Sediment, Seagrass and Dugongs (Dugong dugon)*. Ph.D. thesis, University of Queensland. 2.3.1, 7.1

HEAPS, N. AND GREENBERG, D. (1974). Mathematical model studies of tidal behaviour in the Bay of Fundy. *Oceans*, **6**, 388–399. 2.3

Heaps, N. and Jones, J. (1976). The distribution of the M2 ocean tide in the vicinity of the Bay of Fundy – from a global numerical model by W. Zahel. *Institute of Oceanographic Sciences Report 25*, 1–18. 2.3

Heathcote, R. (1974). *Drought in Australia: a problem of perception*, chap. 3, 103–115. Sorrett Publishing. 2.4.2

Hirsch, C. (1984). *Numerical computation of internal and external flows*, vol. 2. Wiley-Interscience. 2.2, 2.2.3, 2.2.3, 4.1.4, 4.1.4

Hodge, J., Longstaff, B., Steven, A., Thornton, P., Ellis, P. and McKelvie, I. (2005). Rapid underway profiling of water quality in Queensland estuaries. *Marine Pollution Bulletin*, **51**, 113–118. 1.1, 2.3.1, 2.4.3, 7.1

Hossain, S., Eyre, B. and McKee, L. (2004). Impacts of dredging on dry season suspended sediment concentration in the Brisbane river estuary, Queensland, Australia. *Estuarine, Coastal and Shelf Science*, **61**, 539–545. 7.3.2

Howes, P. (2006). A decision support-system for protecting water quality in Moreton Bay: draft users guide to DSS components. *Moreton Bay Waterways and Catchments partnership, Brisbane, Australia*, 12pp. 2.4.3

Hubbard, M. (1999). Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids. *J. of Comp. Phys.*, **155**, 54–74. 4.3

ICSM (2006). Geocentric datum of Australia technical manual. Tech. rep., Intergovernmental committee on surveying and mapping. 7.2.2

Jha, A. (2006). *Finite Volume Hydrodynamic Modeling using Unstructured Meshes*. Ph.D. thesis, Loughborough University, Department of Civil and Building Engineering. 1.2, 2.1.1, 2.2.3, 3.1.3, 4, 4.1.3, 4.1.3, 4.1.4, 4.1.4, 4.8, 4.3.3, 5, 5.3.3, 5.6

Kachiashvili, K., Gordeziani, D., Lazarov, R. and Melikdzhanian, D. (2007). Modeling and simulation of pollutants transported in rivers. *Appl. Math. Modelling*, **31**, 1371–1396. 2.3.1, 4.3.4

KARELSKY, K. AND PETROSYAN, A. (2006). Particular solutions and Riemann problem for modified shallow water equations. *Fluid Dynamics Research*, **38**, 339–358. 3.1.2, 3.1.3

KAY, A. (2003). On the meaning of "Object-Oriented Programming". www.purl.org/stefan_ram/pub/doc_kay_oop_en, webpage hosted by Stefan Ram. 4.2

KRANENBURG, C. (1992). Wind-driven chaotic advection in a shallow model lake. *J. of Hydr. Res.*, **30**, 29–46. 2.1.2, 5.4, 5.5, 8.1.3

LABROSSE, J., NOERGAARD, T., OSHANA, R., WALLS, C. AND CURTIS, K. (2007). *Embedded software*. Elsevier, illustrated edn. 1

LANE-SERFF, G. AND WOODWARD, M. (2001). Internal bores in two-layer exchange flows over sills. *Deep-Sea Research Part I*, **48**, 63–78. 3.2.7

LARSEN, G. (2007). *Modelling hydrodynamic processes within Pumicestone Passage, northern Moreton Bay, Queensland*. Ph.D. thesis, Queensland University of Technology. 2.4.3, 6.2, 6.2.2, 6.3, 7.3.2, 8.1.5

LAWRENCE, G. (1981). On the hydraulics of Boussinesq and non-Boussinesq two-layer flows. *J. Fluid Mechanics*, **215**, 457–480. C.2.1

LEVEQUE, R. (1990). *Numerical methods for conservation laws*. Birkhäuser. 4.1.3

LEVEQUE, R. (2003). *Finite Volume Methods for Hyperbolic Problems*. CUP. 2.2, 3.1, 3.1, 3.1.1, 3.1.2, 3.1.3, 4.1.1, 4.1, 4.1.2

LI, J. AND CHEN, G. (2006). The generalized Riemann problem method for the shallow water equations with bottom topography. *Int. J. for Num. Meth in Eng.*, **65**, 834–862. 3.1.2, 3.1.3

LIANG, D., FALCONER, R. AND LIN, B. (2006). Improved numerical modelling of estuarine flows. *Maritime Engineering*, **159**, 25–35. 2.1.1, 2.3, 4.3.5, 7.3.1

LIANG, Q. AND BORTHWICK, A. (2009). Adaptive quadtree simulations of shallow flows with wet-dry fronts over complex topography. *Computers and Fluids*. 2.3

LIANG, Q., TAYLOR, P. AND BORTHWICK, A. (2009). Particle mixing and reactive front motions in chaotic but closed shallow flows. *Computers and Fluids*. 2.3.1

Lin, B. and Falconer, R. (1997). Tidal flow and transport modeling using ULTIMATE QUICKEST scheme. *J. of Hydr. Eng.*, 303–314. 2.3.1, 2.3.1, 4.3.4

Lin, G., Lai, J. and Guo, W. (2003). Finite-volume component-wise TVD schemes for 2D shallow water equations. *Advances in Water Resources*, **26**, 861–873. 2.2.6, 2.3

Liu, J., Lin, I., Shih, R., M.Z. Chen and Hsieh, M. (1996). Object-oriented programming of adaptive finite element and finite volume methods. *Appl. Num. Math.*, **21**, 439–467. 2.5

Ludwig, R., Mauser, W., Niemeyer, S., Colgan, A., Stolz, R., Escher-Vetter, H., Kuhn, M., Reichstein, M., Tenhunen, J., Kraus, A., Ludwig, M., Barth, M. and Hennicker, R. (2003). Web-based modelling of energy, water and matter fluxes to support decision making in mesoscale catchments – the integrative perspective of GLOWA-Danube. *Phys of Chem. of the Earth*, **28**, 621–634. 2.5.1

Martinez, C., Campbell, K., Annable, M. and Kiker, G. (2008). An object-oriented hydrologic model for humid shallow water-table environments. *J. of Hydr.*, **351**, 363–381. 2.5.1

Mete Uz, B., Donelan, M., Hara, T. and Bock, E. (2002). Laboratory studies of wind stress over surface waves. *Boundary-layer Meteorology*, **102**, 301–331. 2.1.1

Milewski, P., Tabak, E., Turner, C., Rosales, R. and Menazquez, F. (2004). Non-linear stability of two-layer flows. *Comm. Math. Sci.*, **2**, 427–442. 3.1

Muñoz-Ruiz, M., Castro-Díaz, M. and Parès, C. (2000). On a one-dimensional bi-layer shallow-water problem. *Nonlinear analysis*, **53**, 567–600. C.2.1

Namin, M., Lin, B. and Falconer, R. (2004). Modelling estuarine and coastal flows using an unstructured triangular finite volume algorithm. *Advances in Water Resources*, **27**, 1179–1197. 2.2.4, 2.2.6, 2.3, 4.1.2, 4.1.2, 4.1.5, 4.3.5

Neill, S., Elliott, A. and Hashemi, M. (2008). A model of inter-annual variability in beach levels. *Continental Shelf Res.*, **28**, 1769–1781. 2.3.1, 2.3.1, 7.4.2, 8.1.4

Newham, L., Letcher, R., Jakeman, A. and Kobayashi, T. (2004). A framework for integrated hydrologic sediment and nutrient export modelling for catchment-scale management. *Env. Modelling and Software*, **19**, 1029–1038. 2.5.1

Ockendon, H. and Ockendon, J. (2004). *Waves and compressible flow*. Springer. 3.1.4

Ockendon, J., Howison, S., Lacey, A. and Movchan, A. (1999). *Applied Partial Differential Equations*. O.U.P., 2nd edn. 2.1, 2.2, 3.1, 3.1.2, 3.1.2, 3.1.3, 3.1.4

Olley, J., Wilkinson, S., Caitcheon, G. and Read, A. (2006). Protecting Moreton Bay: how can we reduce sediment and nutrients load by 50%. Tech. rep., CSIRO Land and Water. 2.4, 7.1

Pedlosky, J. (2003). *Waves in the Ocean and Atmosphere: Introduction to Wave Dynamics*. Springer. 3.2.4

Periáñez (2004). A particle-tracking model for simulating pollutant dispersion in the Strait of Gibraltar. *Marine Pollution Bulletin*. 1.1, 2.3.1, 4.3.4, 6.1.3

Rajsbaum, S. and Viso, E. (2005). Object-oriented algorithm analysis and design. *Sci. of Comp. Prog.*, **54**, 25–47. 2.5

Rasmussen, H. and Badr, H. (1979). Validation of numerical models of the unsteady flow in lakes. *Appl. Math. Modelling*. 2.3

Ray, A. and Bhattacharjee, J. (2005). Time-dependent study of the circular hydraulic jump using the shallow-water approach. *arXiv.org:cond-mat*, **2**. 2.1.2

Roe, P. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comp. Phys.*, **43**, 357–372. 2.2.3, 2.2.4, 3.1.3, 4.1.1, 4.1.1

Salmon, R. (2002). Numerical solution of the two-layer shallow water equations with bottom topography. *J. of Marine Research*, **60**, 605–638. C.2.1

Shanahan, P. (2001). *River Water Quality Model No. 1*, vol. Introduction. IWA publishing. 2.4.3

Slim, A. and Huppert, H. (2004). Self-similar solutions of the axisymmetric shallow- water equations governing converging inviscid gravity currents. *J. Fluid Mech.*, **506**, 331–355. 2.1.2

Spanou, M. and Chen, D. (2000). An object-oriented tool or the control of point-source pollution in river systems. *Env. Modelling and Software*, **15**, 35–54. 2.5.1

Stakgold, I. (1979). *Green's functions and boundary value problems*. Wiley-Interscience. 2

Stoker, J. (1957). *Water waves*. Interscience, 3rd edn. 2.1

Sweby, P. (1984). High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Nuer. Anal.*, **21**, 995–1011. 2.2.3, 4.1.4

Tabak, E. (2007). PDEs notes, NYU. http://www.math.nyu.edu/faculty/tabak/PDEs/CL.pdf. 3.1.2

Toro, E. (1999). *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer, 2nd edn. 3.1.3

Trenberth, K., Large, W. and Olson, J. (1989). The effective drag coefficient for evaluating wind stress over the oceans. *Amer. Met. Soc.*, **2**, 1507–1516. 2.1.1, 3.2.4

Van Leer, B. (1979). Towards the ultimate conservative difference scheme V. a second-order sequel to Godunov's method. *J. Comput. Phys.*, **32**, 101–136. 2.2.3

Vázquez-Cendón, M. (1999). Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry. *J. of Comp. Phys.*, **148**, 497–526. 2.2.3, 3.2.6, 4.1.3, 5.1.1, 8.1

Versteeg, H. and Malalasekera, W. (1995). *An introduction to computational fluid dynamics: the finite volume method*. Longman. 4.1.3

von Neumann, J. and Richtmeyer, R. (1950). A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, **21**, 232–237. 2.2, 2.2.3

Wallbrink, P. (2004). Quantifying the erosion processes and land-uses which dominate fine sediment supply to Moreton Bay, Southeast Queensland, Australia. *J. of Environmental Radioactivity*, **76**, 67–80. 1.3, 2.4.1, 2.4.3

Wang, J., He, Y. and Ni, H. (2003). Two-dimensional free surface flow in branch channels by a finite-volume TVD scheme. *Advances in Water Resources*, **26**, 623–633. 2.2.6, 2.3

Whitham, G. (1974). *Linear and nonlinear waves*. Wiley-Interscience. 2.1, 3.1.3, 5.3.1

Whitmore, M. and DeLacy, T. (2004). Sustainable development and management of tourism in Moreton Bay. *Brisbane City Council*. 7.4.2

Yee, H., Warming, R. and Harten, A. (1985). Implicit total variation diminishing (TVD) schemes for steady-state calculations. *J. of Comp. Phys.*, **57**, 327–360. 2.2.3

Yoon, T. and Kang, S. (2004). Finite volume model for two-dimensional shallow water flows on unstructured grids. *Journal of Hydraulic Engineering*, 678–688. 2.2.3, 2.2.6, 2.3

You, Z. (2005a). Estimation of bed roughness from mean velocities measured at two levels near the seabed. *Continental Shelf Research*, **25**, 1043–1051. 2.4.3

You, Z. (2005b). Fine sediment resuspension dynamics in a large semi-enclosed bay. *Ocean Engineering*, **32**, 1982–1993. (document), 2.3.1, 2.4.3, 7.1, 7.2.1, 7.3, 7.3.1, 7.3.1, 7.3.1, 7.4, 7.5, 7.6, 7.4.2, 7.5, 8.1.4, 8.1.5

Yu, L. and O'Brien, J. (1991). Variational estimation of the wind stress drag coefficient and the oceanic eddy viscosity profile. *J. Physical Oceanography*, **21**, 709–719. 2.1.1, 3.2.4

Zhou, J., Causon, D., Mingham, C. and Ingram, D. (2001). The surface gradient method for the treatment of source terms in the shallow water equations. *J. of Comp. Phys.*, **168**, 1–25. 2.3