The open boundary equation
Master Thesis
by
Dirk Diederen

in partial fulfillment of the requirements for the degree of

Master of Science
in Hydrology (Watermanagement, Civil engineering),

at the Delft University of Technology,
to be defended publicly on Thursday June 11, 2015 at 16:00 hr.

Supervisor: Prof. H.H.G. Savenije, TU Delft
Thesis committee: Dr. M. Toffolon, University of Trento
Dr. S. Pande, TU Delft
Prof. W.S.J. Uijttewaal, TU Delft
Prof. A.J.H.M. Reniers, TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.
# Contents

1 **Introduction** .................................................. 1
   1.1 Idea ......................................................... 2
   1.2 Report setup .................................................. 2
   1.3 Background ................................................... 3
   1.4 Governing Equations ............................................ 5
      1.4.1 Eulerian Perspective ...................................... 5
      1.4.2 Riemann Perspective ...................................... 5

2 **Analytical Work** ............................................... 7
   2.1 Formal Derivation in a ‘Perfect’ Channel .................. 8
   2.2 Analytical Approximations .................................. 10
      2.2.1 Description .............................................. 10
      2.2.2 Progressive Wave without Damping ..................... 10
      2.2.3 Mixed Wave with Damping ............................... 11

3 **Numerical Work** ............................................... 13
   3.1 Converging channels with friction ............................. 14
   3.2 Why Numerical Approximations? ............................... 14
   3.3 Model Toffolon ............................................... 15
      3.3.1 Description .............................................. 15
      3.3.2 Simulations .............................................. 16
      3.3.3 Results .................................................. 17

4 **Interpretation** ................................................. 21
   4.1 Simulating an Open Boundary Condition .................... 22
   4.2 Domain-splitting and Internal Open Boundaries .......... 24

5 **Wave character** ............................................... 27
   5.1 Phase Lag and Reflection .................................... 28
   5.2 Lagrangean Analysis ......................................... 28
   5.3 Apparent vs. Classical Reflection .......................... 30

6 **Applications** .................................................. 31
   6.1 Reconstructing the Velocity .................................. 32
   6.2 Estimation of Friction ....................................... 34

7 **Conclusions** .................................................. 37
   7.1 Summary ...................................................... 38
   7.2 Further Research ............................................. 39
A  Process ........................................................................................................ 41
   A.1  General Description ........................................................................ 42
   A.2  Additional thesis ............................................................................ 42
   A.3  Master thesis ................................................................................ 42
B  Tools ........................................................................................................ 43
   B.1  Numerical Model Toffolon ............................................................. 44
       B.1.1  Governing Equations ............................................................ 44
       B.1.2  Numerics ............................................................................ 44
       B.1.3  Acknowledgments ............................................................... 44
   B.2  MATLAB ....................................................................................... 45
       B.2.1  analysisDirk ...................................................................... 45
       B.2.2  picLagrCorrWidth .............................................................. 57
C  Additional Simulations ........................................................................... 59
   C.1  Time series .................................................................................... 59
D  Literature ................................................................................................ 63
   References ........................................................................................... 63
Figure 1.1: The problem.

Fairy tales are more than true: not because they tell us that dragons exist, but because they tell us that dragons can be beaten.

G.K. Chesterton
1.1. Idea
Consider an estuary which is described by two governing equations for two variables, the water level $\zeta$ and the velocity $u$. Having two governing equations is sufficient to perform time integration.

The problem statement for a finite domain contains two boundary conditions and an initial condition. However, if the domain is infinitely long, we only have one boundary condition. Does this mean that the problem statement for an infinite domain contains less information?

The idea is that the loss of one boundary condition, by assuming an infinite domain, leads to an addition equation, containing the latent information that the domain is infinitely long. From two equations with two boundary conditions, we obtain three equations and one boundary condition to formulate the problem statement. This idea is further elaborated in Chapt. 4.

Therefore, we call this additional equation ‘the open boundary equation’,

$$\zeta_t u_x = u_t \zeta_x,$$

where the subscripts refer to partial derivatives.

1.2. Report setup
1. In this report we consider a long, exponentially converging estuary, where the influence of the landward boundary condition is negligible so that the reflected wave is negligible. This is quite a common case in real alluvial estuaries, and can be approximated as an ‘infinitely long’ channel, where the reflected wave is non-existent. Here we derive a new, additional equation, which appears to be a part of the problem formulation.

2. First we derive the equation exactly in a frictionless, prismatic, horizontal channel in Sect. 2.1,

3. then we test the performance of the equation in numerical simulations with friction and convergence in Sect. 3.3.3.

4. We discuss the origin of the new equation in Chap. 4.

5. Then we present a new method to look at the tidal wave character in Chap. 5.

6. Finally, we propose applications of the new equation for hydrological and hydraulic purposes in Chap. 6.
1.3. Background

The behavior of tidal waves in natural estuaries has been the subject of numerous studies with different degrees of complexity. For rapid assessment, the tidal dynamics in these estuaries may be described as one-dimensional, whereby the standard Saint-Venant shallow water equations are used as the governing equations. For long waves, one-dimensional descriptions are appropriate in long water bodies, where the length is much larger than the width.

De Saint-Venant [4] formulated the governing equations in the Eulerian reference frame, which has the advantage that the reference frame does not move. An exact analytical solution of such equations is not available, so various approximations have been proposed. Perturbation methods [e.g., 6–8] are particularly useful to this aim, since they can capture the approximate dynamics conceptually. The Eulerian reference frame is also convenient for numerical methods, where a grid has to be defined and a non-moving grid has a clear advantage. These numerical approximations may not provide conceptual relations between parameters, but they do permit the analysis of complex geometries and retain the full nonlinear behavior. In addition to the Eulerian perspective, we may rewrite the governing equations in the Lagrangean reference frame, where an observer moves with the speed of the water particle. This provides additional insight in the wave character and allows for a different approximate analytical approach [13, 14]. There the structure of the solution is assumed a priori in the form of a sinusoidal function, but where nonlinear terms in advection and friction can be retained. A third perspective is that of the ‘Riemann invariants’ [11], or ‘method of characteristics’, where the observer moves with the propagation speed of disturbances. For long estuaries, this ‘celerity’ is a main parameter in most analytical approximate solutions. Moreover, the Riemann perspective provides approximations for open boundaries.

Analytical descriptions can easily treat infinitely long channels, whereas numerical models require an artificial ‘open’ boundary condition [10], where the propagating wave exits a limited domain unaffected. Such a boundary can also be referred to as ‘absorbing’ [5], since waves that hit this boundary are absorbed instead of reflected (as would be the case when they hit a closed boundary), or as ‘non-reflective’ [9]. Perfectly functioning open boundaries in numerical tidal hydraulics have yet to be defined, so a small part of the wave is always reflected. Yet, a perfect open boundary condition is required for long estuaries where the examined domain is limited. Using an open boundary, the domain can be split into multiple reaches to be solved subsequently [e.g., 9, 16]. Theoretically, with this splitting method the domain can be divided into an infinite number of reaches. Since the open boundary condition is applied at each location in the domain, this condition becomes an equation. Here we present the ‘open boundary equation’.

The Riemann invariants suggest that any solution for sub-critical flow is composed of pieces of information that are advected in positive and negative direction. This also applies for a wave which physically seems to move only in one direction. However, the tidal wave character is usually described as either a single incident wave or the superposition of an incident and a reflected wave. This superposition finds its origin in linear wave theory, specifically for frictionless, prismatic, horizontal
channels with a closed landward boundary, where the incident wave and its reflection have the same strength. In that basic scenario a phase lag of 90 degrees can be observed between the water depth and the velocity. Interestingly, a phase lag can also be observed in long, converging estuaries, where there is no such closed boundary. The terminology to describe the tidal wave character in such natural estuaries appears to be subject to debate. Is the phase lag, which is observed in long converging estuaries, similar to the phase lag that is caused by the reflection of a closed landward boundary? In other words, should we call it ‘reflection’? Hunt [7] warned that the use of the classical terms ‘progressive wave’ and ‘standing wave’ to a frictional situation should be treated with ‘considerable caution’. He prefers the description by a single wave over the description by a superimposed incident and reflected wave, and he calls this single wave standing when it shows a phase lag. Jay [8] argued that the use of descriptions like ‘progressive’ and ‘standing’ should not be used in convergent estuaries, for two reasons. First, friction and convergence play a large role in estuaries and so even perfect reflection by a barrier would lead to a standing wave only at the point of reflection. Second, any phase lag from 0 to 90 degrees can occur for a single incident wave. Friedrichs and Aubrey [6] found that a single incident wave in a converging estuary can have a 90 degrees phase lag, but can have a celerity close to that of a frictionless progressive wave. Savenije et al. [15] described that a wave with a mixed character occurs in convergent estuaries and attributes this to convergence and friction. They also showed that the wave character depends on the phase speed, which in turn depends on the phase lag. Nevertheless, all these authors concluded that the ‘apparent reflection’ in an estuary is different from the ‘classical reflection’ produced by a closed boundary.
1.4. Governing Equations

1.4.1. Eulerian Perspective
The Saint-Venant equations are generally used to analyze water flow in channels and estuaries. Together the mass balance (1.2) and momentum balance (1.3) describe the movement of the water, where, within the Eulerian perspective, the subscripts refer to partial derivatives in space and time. These equations are useful for an observer at a given location along the channel,

\[ h_t + uh_x + hu_x - \beta uh = 0, \]  
\[ u_t + uu_x + g\zeta_x + W = 0, \]  
\[ h = \zeta - Z, \quad \beta = -\frac{B_x}{B} \]  

where

- \( x \) is the longitudinal coordinate (positive landward),
- \( t \) is the time,
- \( g \) is the acceleration due to gravity,
- \( u(x,t) \) is the cross-sectional average flow velocity,
- \( h(x,t) \) is the water depth,
- \( \zeta(x,t) \) is the water level,
- \( Z(x) \) is the bed level,
- \( B(x) \) is the width and
- \( W(x,t) \) is the frictional term.

1.4.2. Riemann Perspective
The observer can also choose to move with the wave, for which rewriting Eqs. (1.2) and (1.3) into the Riemann perspective of invariants is convenient,

\[ R_{1,t} + (u + c)R_{1,x} = \beta uc - gZ_x - W, \]  
\[ R_{2,t} + (u - c)R_{2,x} = -\beta uc - gZ_x - W, \]  

where the Riemann invariants are defined by

\[ R_1 = u + 2c, \quad R_2 = u - 2c, \]  

and \( c = \sqrt{gh} \) is the long wave celerity for small amplitudes.

In this report we restrict our focus on sub-critical flow since this is the typical case in tidal flows. Hence, the Froude number \( |u|/c < 1 \), so that one invariant is advected in the positive direction and the other in the negative direction, as can be seen in Eqs. (1.5) and (1.6).
An analytical solution to a problem is one that has a "proof": a series of logical steps that can be followed and verified as correct. If you use the quadratic formula to solve for $x$ in a quadratic equation, this is an analytical solution to the problem.

Alvin Grissom II
2.1. Formal Derivation in a ‘Perfect’ Channel.

The first step towards the open boundary equation is to derive it analytically for a ‘perfect channel’, which we specify as an infinitely long, prismatic channel where convergence and friction are zero and the bed is horizontal, hence $B_x = 0$, $W = 0$ and $h_x = \zeta_x$. The basic equations in Eulerian form (1.2) and (1.3), and in Riemann form (1.5) and (1.6) then reduce to:

\[
\begin{align*}
    h_t + uh_x + hu_{xx} &= 0, \\
    u_t + uu_{xx} + gh_x &= 0, \\
    R_{1,t} + (u + c)R_{1,x} &= 0, \\
    R_{2,t} + (u - c)R_{2,x} &= 0.
\end{align*}
\]

(2.1)
(2.2)
(2.3)
(2.4)

We assume an infinite unbounded domain from $x = 0$ to $x = \infty$ to avoid reflection from a closed landward boundary. We put a forcing boundary condition seaward. As initial condition, we impose a constant $R_2(x,0) = C$, where starting with a constant velocity (e.g. $u(x,0) = 0$) and a constant water level (e.g. $\zeta(x,0) = 0$) is the easiest option.

Figure 2.2: In an infinite domain, the trajectory of the negative invariant intersects the initial condition. Along its trajectory, its value does not change.
2.1. Formal Derivation in a ‘Perfect’ Channel.

As can be observed in Fig. 2.2, tracking the seaward propagating invariant $R_2(\hat{x}, \hat{t})$ back in time from any location $\hat{x}$ within the domain, eventually the trajectory will lead back to an initial condition $R_2(\hat{x}, 0)$. This arrival point $x_1$ will also be in the infinite domain, because in sub-critical flow the celerity $(u - c)$ of $R_2$ is negative, so that $\hat{x} > \hat{x}$.

If we integrate Eq. (2.4) along the pathway of $R_2$ from $\hat{x}$ to $\hat{x}$,

$$R_2(\hat{x}, \hat{t}) = R_2(\hat{x}, 0) + \int_0^t \frac{dR_2}{dt} dt,$$

(2.5)

where the operator $d/dt = \partial/\partial t + (u - c) \partial/\partial x$ represents the full Riemann derivative for $R_2$, we find that $R_2(\hat{x}, \hat{t}) = R_2(\hat{x}, 0)$ (hence the term invariant).

Since $\hat{x}$ can be any location in the domain and $\hat{t}$ can be any time, the initial condition $R_2(x, 0) = C$ in an infinitely long channel with sub-critical flow leads to $R_2(x, t) = C$.

This means that if we differentiate Eq. (1.7b) in space and in time, we obtain

$$R_{2,x} = 0,$$

(2.6)

$$R_{2,t} = 0,$$

(2.7)

so that

$$\sqrt{h} u_x = \sqrt{gh}_x,$$

(2.8)

$$\sqrt{h} u_t = \sqrt{gh}_t.$$  

(2.9)

By combining Eq. (2.8) with Eq. (2.9) we obtain

$$h_t u_x = u_t h_x,$$

(2.10)

and

$$hu_t u_x = gh_t h_x.$$  

(2.11)

Moreover, from the Lagrangean perspective we find

$$\frac{Dh}{Dt} u_x = \frac{Du}{Dt} h_x,$$

(2.12)

$$\sqrt{h} \frac{Du}{Dt} = \sqrt{gh} \frac{Dh}{Dt},$$

(2.13)

where the operator $D/Dt = \partial/\partial t + u \partial/\partial x$ represents the full Lagrangean derivative. From the Riemann perspective we find

$$R_{1,t} R_{2,x} = R_{2,t} R_{1,x},$$

(2.14)

$$R_{1,t} R_{2,x} = -R_{2,t} R_{1,x}.$$  

(2.15)

which seem to indicate a contradiction. However, this is not the case since one of the invariants is constant in space and time, so the contradiction is removed by triviality, as zero is zero.
These equations all contain the same latent information for a ‘perfect channel’, i.e. if one applies, so do the others. What is enclosed in these equations is the infinite domain, which implies they are also valid for a simulation with a finite domain and an open landward boundary condition. We further discuss this in Sect. 4.

Since for a horizontal bed $\zeta_x = h_x$, we rewrite Eq. (2.10) to

$$\zeta_t u_x = u_t \zeta_x, \quad (2.16)$$

which we call ‘the open boundary equation’. This way the latent information is expressed independent of the geometry. Examples to derive Eq. (1.1) from simple analytical approximations are presented in Appendix 2.2. Could it be that Eq. (1.1) also applies more generally, in estuaries with friction, convergence and a bed slope?

### 2.2. Analytical Approximations

#### 2.2.1. Description

In this section we consider some analytical approximations of progressive wave solutions [e.g., 14], to verify if Eq. (1.1) holds, and under which conditions. $\Omega$ and $\Psi$ are any periodic function, $\eta$ and $\nu$ are amplitudes of water level and velocity, $\varphi$ is the variable part of the argument which is the same for $\zeta$ and $u$, $\epsilon$ is the phase lag.

#### 2.2.2. Progressive Wave without Damping

A progressive wave can be described by two generic periodic functions as

$$\zeta(x, t) = \eta \Psi(\varphi(x, t)), \quad (2.17)$$

$$u(x, t) = \nu \Omega(\varphi(x, t)), \quad (2.18)$$

whose partial derivatives are

$$\zeta_t = \eta \Psi \varphi_t, \quad \zeta_x = \eta \Psi \varphi_x, \quad (2.19)$$

$$u_t = \nu \Omega \varphi_t, \quad u_x = \nu \Omega \varphi_x.$$  

By multiplying the crossed partial derivatives it is straightforward to obtain that

$$\zeta_t u_x = u_t \zeta_x, \quad (2.20)$$

thus demonstrating the validity of Eq. (1.1) in this case.
2.2.3. Mixed Wave with Damping

In a damped wave, the amplitudes change along $x$. Considering a generic wave allowing for tidal damping/amplification, friction and convergence:

\[ \zeta(x, t) = \eta(x)\Psi(\varphi(x, t) + \epsilon(x)), \quad (2.21) \]
\[ u(x, t) = v(x)\Omega(\varphi(x, t)). \quad (2.22) \]

The derivatives now read

\[ \zeta_t = \eta\Psi_\varphi\varphi_t, \quad \zeta_x = \eta\Psi_\varphi (\varphi_x + \epsilon_x) + \eta_x\Psi, \]
\[ u_t = v\Omega_\varphi\varphi_t, \quad u_x = v\Omega_\varphi\varphi_x + \nu_x\Omega. \quad (2.23) \]

Multiplying the derivatives as in Eq. (1.1) yields more complex results,

\[ \zeta_t u_x = \eta\Psi_\varphi\varphi_t v\Omega_\varphi\varphi_x + \eta\Psi_\varphi\varphi_t\nu_x\Omega, \quad (2.24) \]
\[ u_t \zeta_x = v\Omega_\varphi\varphi_t \eta\Psi_\varphi (\varphi_x + \epsilon_x) + v\Omega_\varphi\varphi_t \nu_x\Psi. \quad (2.25) \]

The validity of Eq. (1.1) in this case requires two conditions,

\[ \epsilon_x = 0, \quad (2.26) \]
\[ \frac{1}{\eta} \frac{\nu_x}{\Omega} \Psi_\varphi = \frac{1}{\nu_x} \frac{1}{\Psi} \Psi_\varphi. \quad (2.27) \]

The first condition (2.26) is valid for a purely progressive wave ($\epsilon = 0$), but also for a mixed wave for which the phase between water level and velocity remains constant along $x$, which is the case in an ideal estuary but also in the asymptotic situation of alluvial estuaries (see Cai and Savenije [1]).

The second condition requires more attention. If the structure of the two waves is characterized by a proportionality, i.e. $\Omega = k\Psi$, as it is often assumed, then (2.27) becomes

\[ \frac{1}{\eta} \frac{\nu_x}{\Omega} = \frac{1}{\nu_x}, \quad (2.28) \]

which represents a requirement on the degree of damping of the velocity and water level amplitudes. In the case of linearized waves, this can be demonstrated to hold for infinitely long channels [16], but the assumption of proportional damping (2.28) is also at the basis of Savenije’s approach [14], which has proved to yield satisfactory results for nonlinear cases [2, 3]. Finally, we note that a mixed wave in an ideal estuary clearly satisfies Eq. (1.1), since by definition in an ideal estuary the phase lag $\epsilon$ does not change (2.26) and no damping is present (2.27).
In contrast to analytical solutions, some problems are solved via other means. If, instead of using the quadratic formula, you try a lot of values for $x$ and $y$, this is a numerical solution. Some problems simply do not have analytical solutions and must be approximated using numerical methods – for example, many complicated integrals.

Alvin Grissom II
3. Numerical Work

3.1. Converging channels with friction
So what happens with the open boundary equation (1.1) when there is convergence of width, a bed slope and friction? These terms do not appear in the open boundary equation, so might it still hold?

Combining the mass balance,

\[ h_t + uh_x + hu_x - \beta uh = 0, \tag{3.1} \]

the momentum balance,

\[ u_t + uu_x + g\zeta_x + W = 0, \tag{3.2} \]

and the open boundary equation,

\[ \zeta_t u_x = u_t \zeta_x, \tag{3.3} \]

yields the following two equations,

\[ hu_x^2 - (Z_x + \beta h) uu_x = g\zeta_x^2 + W\zeta_x, \tag{3.4} \]
\[ hu_t^2 + (hW + \beta hu^2 + u^2Z_x) u_t = gh_t^2 - (W + \beta gh + gZ_x) uh_t. \tag{3.5} \]

Since only substitution has been applied, the following statements can be made:

- if Eqs. 3.1, 3.2 and 3.3 apply, then Eqs. 3.4 and 3.5 also apply.
- if Eqs. 3.1, 3.2 and 3.4 apply, then Eqs. 3.3 and 3.5 also apply.
- if Eqs. 3.1, 3.2 and 3.5 apply, then Eqs. 3.3 and 3.4 also apply.

3.2. Why Numerical Approximations?
As Eqs. 3.4 and 3.5 cannot be simply integrated, it is not possible to check for correctness analytically as in Sect. 2.1, so we have to resort to numerical testing. Numerical results cannot prove analytical equations to be correct, as they are single realizations that cannot be used to extrapolate general results. Moreover, they are approximations that contain numerical errors. However, the validity of a given relationship can be tested by running a large number of numerical simulations: if the results are not contradicting it, the argument supporting its validity becomes stronger as the range of the conditions tested becomes wider.

Therefore, we use numerical experiments to estimate the difference between the left hand and the right hand sides of the equations, and check if they differ within acceptable thresholds. If so, they can be tested for practical applications. Moreover, if the agreement appears to be consistent, further attempts to find formal proof could also be encouraged.
3.3. Model Toffolon

3.3.1. Description

Simulations are performed by solving Eqs. (1.2) and (1.3). The length of the domain $L$ is chosen as twice the wavelength $L_0 = c_0 T$, where $c_0 = \sqrt{g h_0}$ is the linearized tidal wave celerity in 'perfect channels' and $T$ is the tidal period. A sinusoidal periodic oscillation of the water level is imposed at the seaward boundary with an amplitude $\eta$ and a period $T = 12.41 \, h$ corresponding to a dominant $M_2$ tide. At the landward boundary an approximate open boundary condition is applied.

The following width is applied in this model,

$$B = B_\infty + (B_0 - B_\infty) \exp\left(-\frac{x}{b}\right),$$

where $B_0$ is the width at the mouth of the estuary, $B_\infty$ is the asymptotic value of the width and $b$ is the convergence length of the width. The width at the mouth $B_0$ is fixed at 10 km and the asymptotic value $B_\infty$ is fixed at 10 m.

Additionally, an exponentially converging bed level is assumed, with

$$Z = -\bar{h}_0 \exp\left(-\frac{x}{d}\right).$$

where $\bar{h}_0$ is the average depth at the mouth and $d$ is the convergence length of the bed. Since there is a bed slope: $h_x = \zeta_x - Z_x$.

Strickler friction is applied,

$$W = g \frac{u|u|}{K^2 h^{4/3}},$$

where $K$ is the Strickler coefficient.

Starting from the initial conditions $\zeta(x,0) = 0$ and $u(x,0) = 0$, the simulations last until an approximate periodic equilibrium is reached.

For additional information on this numerical model, see Appendix B.1.2.
3.3.2. Simulations
Several 10’000s of simulations have been performed with this model. The following parameter ranges have been explored,

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Unit</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h_0 )</td>
<td>m</td>
<td>2 – 40</td>
<td>Avg depth at mouth</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>-</td>
<td>0.01 – 0.6</td>
<td>Amplitude/depth</td>
</tr>
<tr>
<td>( \gamma_b )</td>
<td>-</td>
<td>0 – 5 (^a)</td>
<td>Width convergence</td>
</tr>
<tr>
<td>( \gamma_d )</td>
<td>-</td>
<td>0 – 0.5 (^b)</td>
<td>Depth convergence</td>
</tr>
<tr>
<td>( K )</td>
<td>m(^{1/3})s(^{-1})</td>
<td>20 – 80</td>
<td>Friction constant</td>
</tr>
</tbody>
</table>

\(^a,^b\) Approximately 15% of the cases the value 0 is chosen.

Table 3.1: Parameter ranges.

in which the dimensionless parameters are

\[ \alpha = \frac{\eta}{h_0}, \quad \gamma_b = \frac{c_0}{b \omega}, \quad \gamma_d = \frac{c_0}{d \omega}, \quad (3.9) \]

where

• \( \eta \) is the tidal amplitude at the mouth (M2),

• \( \omega = 2\pi T^{-1} \) is the tidal frequency and

• \( c_0 = \sqrt{g h_0} \) is the linearized tidal wave celerity.
3.3. Results

Figure 3.2: Time series are displayed of the terms in Eq. (3.4) at location $x = 0.2L$ for an estuary with $\eta = 0.1 \text{ m}$, $h_0 = 10 \text{ m}$, $b = 70 \text{ km}$, $d = 200 \text{ km}$, $\kappa = 45 \text{ m}^{1/3}\text{s}^{-1}$. In order to find good agreement between the left hand side and the right hand side, none of the terms can be neglected.

The governing equations (3.1) and (3.2) are evaluated by the numerical model, so we can choose to check Eq. (3.3), Eq. (3.4) or Eq. (3.5), as explained in Sect. 3.1. We start with time series analysis of a single simulation for Eq. (3.4) in Fig. 3.2, which contains friction $W$, convergence $\beta$ and bed slope $Z_x$.

$$h u_x^2 - (Z_x + \beta h) uu_x = g \zeta_x^2 + W \zeta_x,$$  \hfill (3.10)

In a single wave period, two peaks occur in the squared terms, where the larger peak is the steeper wave front and the smaller is the rear. The size of these peaks is mostly influenced by the amplitude-to-depth ratio, which in this case has been chosen small. This way it can easily be observed that good agreement of the left hand side of Eq. (3.4) and the right hand side can be found when the terms containing the width convergence, the friction and the bed slope are not negligible in size compared to the larger squared terms.
Figure 3.3: The Pearson coefficient $\rho$ is displayed as a function of space, which correlates time series of the left hand side and the right hand side of Eq. (1.1) for an estuary with $H_0 = 10$ m, $b = 100$ km, $d = 250$ km, $K = 40$ m$^{1/3}$s$^{-1}$, $L = 900$ km and a varying amplitude $\eta$.

Figure 3.3 illustrates the longitudinal variation of the Pearson correlation coefficient $\rho$, which correlates time series of the left hand side of Eq. (1.1) and the right hand side. It suggests that the validity of the proposed equation is asymptotically reached, with a clear effect of the seaward boundary, where a higher amplitude-to-depth ratio starts with a lower correlation, but converges faster.

It should be observed that the imposed seaward boundary is not a naturally occurring relationship between water level and flow, and therefore is not consistent with Eq. (1.1). The disturbance created by the seaward boundary condition requires a certain length for the tidal wave to adjust.

There also is a clear effect of the landward boundary, as a result of the weakly reflective numerical boundary condition. For horizontal estuaries, this effect disappears quickly in seaward direction due to channel convergence and friction, but a bed slope severely increases the length of this effect.
The result of several typical cases is shown in Fig. 3.4. Each point represents a calculation at a certain location and time in the estuary of Eq. (1.1), which has been made dimensionless by dividing by the scale factor $\eta^2 / (h_0 T^2)$ to get points of as many different scenarios as possible in a single figure.

Small values of Eq. (1.1) demonstrate considerable scatter, which may be explained by numerical errors, which are relatively large in this range. Large values give an almost perfect agreement.
Figure 3.5: To summarize the spatial correlations (see Fig. 3.3) of 10,000 simulations with a horizontal bed and with parameter ranges \( \kappa = [0, 0.6], \gamma = [0, 5], \chi = [0, 60] \), three parameters have been explored.

In order to show even more simulations, we have summarized Figure 3.3 by three indicators of the locations where \( \rho \) exceeds a threshold (which we choose equal to 0.98),

1. we show the distance to the first point that exceeds this threshold from the seaward boundary, ‘ic1’.
2. we do the same for the distance from the landward boundary, ‘ic2’.
3. we count the fraction of locations where the threshold is exceeded, ‘nc’.

For a great number of simulations with a horizontal bed, Figure 3.5 shows the results for these three indicators as a function of three dimensionless parameters,

\[
\kappa = \frac{\eta_0}{h_0}, \quad \gamma = \frac{\beta c_0}{\omega}, \quad \chi = \frac{c_0 \kappa}{\omega h_0 C^2}, \quad (3.11)
\]

which includes a dimensionless friction parameter.

Finally, time series of the left hand side of Eq. (1.1) and the right hand side, for a few example scenarios, are presented in Appendix C.1.
Figure 4.1: Huh? What?

Less information often leads to more interpretation.

John Paul Caponigro
4.1. Simulating an Open Boundary Condition

The objective of an ‘open’ or ‘absorbing’ boundary condition in wave theory is to not create reflection. It is appropriately used when posed far enough from a reflecting physical boundary, since otherwise physical reflection should be accounted for. It can be applied in sub-critical flow, where information travels in two directions: positive \((R_1)\) and negative \((R_2)\).

The length of influence of a reflection depends on several parameters, such as friction, convergence, bed slope and amplitude-to-depth ratio. Mathematically, the only way to be certain that there is no reflection is to simulate an infinitely long channel. In fact this means that the perfect open boundary condition implies that behind it there is an infinitely long channel.

A local open boundary should mimic the information that otherwise would have come from outside the domain. This is simple and well known for a ‘perfect channel’, where as initial condition outside the domain \(R_2\) is assumed to be constant and where the value of \(R_2\) does not change along its trajectory (as Eq. (2.4) implies),

\[
R_{2,t} + (u - c)R_{2,x} = 0. \tag{4.1}
\]

However, it is more difficult to reproduce this condition when additional terms like friction, convergence and bed slope are accounted for as in Eq. (1.6), whereby the values of the invariants change along their trajectories,

\[
R_{2,t} + (u - c)R_{2,x} = -\beta u c - gZ_x - W. \tag{4.2}
\]

In other words, then the term ‘invariants’ is no longer appropriate. The changes in value of \(R_2\) along its trajectory and the trajectory itself depend on the local solution of the water depth \(h\) and the velocity \(u\), which cannot be calculated on a part of the trajectory (see the dashed lines in Fig. 4.2), since that part lies outside the domain. Approximations for open boundaries exist for linearized cases [e.g., 5] or with non-local formulas [e.g., 9].
4.1. Simulating an Open Boundary Condition

Figure 4.2: At each point (A, B, C, D) the solution for the water depth $h$ and the velocity $u$ depends on the values of both Riemann invariants. Each trajectory (which is implicitly dependent on the solution) leads back to an initial condition or a boundary condition.

As shown in Fig. 4.2, in sub-critical flow the solution can depend on: (i) two boundary conditions (point A); (ii) one boundary value and one initial condition (B, C); or (iii) two initial conditions (D). An open boundary, which is represented by the dash-dot line, turns point A into the second option, and point B into the third option through the extension by the dashed lines.
4.2. Domain-splitting and Internal Open Boundaries

The simulation of a progressive wave in a computational domain can be divided into sub-domains with internal boundary conditions \([9, 16]\). The sub-domains can be explicitly solved from left to right if the reflected wave does not exist or can be neglected, so that the landward boundary can be seen as open. Hence, the solution at the open boundary of each sub-domain can be used as the seaward boundary condition for the next sub-domain (see Fig. 4.3). Simulation with sub-domains should result in the same physical simulation as when using a single domain, whereby it is not sufficient for the open boundaries to represent infinite length alone, but also the correct properties of the channel, like convergence length, friction coefficient and bed slope.

Let us assume that it is generally possible to define such internal open boundaries for estuaries with friction, convergence and a bed slope (and not merely for a ‘perfect channel’). Each internal open boundary should then contain the same information in the sense that behind the boundary lies an infinitely long channel with the same friction and convergence properties as in the entire domain. This implies that the total domain can be described by an infinite number of subdomains, and therefore by an infinite number of open boundaries (see Fig. 4.3), all containing the same physical information. The boundary condition, which applies locally, becomes an additional equation valid at each location in the domain. This means that instead of an open boundary condition, the additional equation \((1.1)\) is part of the problem formulation for an infinite channel.

Similarly, an additional equation (Eq. \((2.28)\)) was applied by Savenije \([14]\), where he assumed a proportional damping equation to represent the open boundary equation, valid for long open boundary estuaries.
Figure 4.3: The same simulation is displayed on a domain with: (a) a single open boundary; (b) two open boundaries; or (c) a large number (theoretically infinite) of open boundaries.
Figure 5.1: What is the wave character?

*The essential thing is not knowledge, but character.*

Joseph Le Conte
5.1. Phase Lag and Reflection

Descriptions of waves as ‘progressive’ or ‘standing’ originate from the analysis of channels which are frictionless, prismatic and horizontal. The primary indicator for these classifications is the phase lag between the water height $h$ and the velocity $u$, which is $0$ degrees for a perfect progressive wave and $90$ degrees for a perfect standing wave. In Sect. 2.1 such a ‘perfect channel’ was analyzed, with an infinite length and therefore no physical, closed boundary. The absence of reflection allows a progressive wave to occur. If such a channel has a closed boundary, full reflection will cause a standing wave to occur.

Clearly, reflection leads to a phase lag. However, we also expect a phase lag in a converging estuary with an open boundary. Moreover, friction and bed slope will influence the phase lag. If we observe a phase lag, does this mean there is reflection?

5.2. Lagrangean Analysis

When we follow a fixed volume that moves through a horizontal channel with a varying width, it is compressed in both longitudinal and lateral direction. Dictated by the mass balance, the variation of the water depth follows from this compression. To be able to follow a particle with a given volume of water in a converging estuary, Eq. (1.2) is rewritten in the Lagrangean perspective [12],

$$\frac{1}{h} \frac{Dh}{Dt} + u_x - \beta u = 0,$$

(5.1)

This equation shows that the relative change of the depth of a water volume is determined by the velocity gradient and the convergence. When tracking this fixed volume, the displacement $S$ can be obtained by

$$S = \int_{0}^{t} uDt.$$

(5.2)

The general solution of the mass balance (5.1) for a converging estuary is

$$h = h_i \exp \left( \int -u_x Dt \right) \exp \left( \int \beta u Dt \right),$$

(5.3)

where $h_i$ is the initial condition. The first integral represents the longitudinal compression due to the velocity gradient and the second integral the lateral compression due to convergence. Combining Eqs. (5.2) and (5.3), for the case that the convergence degree $\beta$ remains constant ($=b^{-1}$ as for a purely exponential decrease), gives:

$$h = h_i \exp \left( \int -u_x Dt \right) \exp (\beta S).$$

(5.4)

We introduce the auxiliary variable

$$h_p = h \exp (-\beta S),$$

(5.5)
which represents the water depth compensated for the lateral compression due to width convergence. It then follows that the variation of the water depth due to the velocity gradient can be represented as

\[ h_p = h_t \exp \left( \int -u_x \, Dt \right). \]  

(5.6)

This equation can be tested by means of the numerical model. The lateral compression due to the varying width can be removed, using numerical particle tracking for the particle displacement \( S \). After applying Eq. (5.5), the transformed water depth \( h_p \) can be compared to the velocity \( u \).

![Graph showing decomposition of vertical tide separating the effect of convergence by introducing the variable \( h_p \).](image)

Figure 5.2: Decomposition of the vertical tide \( h \) separating the effect of convergence by introducing the variable \( h_p \): (a) Lagrangean time series of \( h \), \( h_p \) and velocity \( v \); (b) the same variables as a function of the displacement. The used scenario is close to an ideal estuary, as \( \eta = 2 \, \text{m}, \, \bar{h}_0 = 10 \, \text{m}, \, b = 70 \, \text{km}, \, d = 0 \, \text{km}, \, K = 45 \, \text{m}^{1/3} \text{s}^{-1} \).

Lagrangean time series of \( h \), \( h_p \) and \( u \) are plotted in Fig. 5.2, together with the Lagrangean tidal ellipse in terms of horizontal and vertical displacements. It can be observed that there is a clear phase lag between \( h \) and \( u \). However \( h_p \) has no phase lag with \( u \), suggesting that \( h_p \) resembles a progressive wave. This resemblance is strongest in an ‘ideal’ estuary, where convergence and friction are balanced so that there is no damping of the amplitude of the tidal wave. So an ‘apparently standing’ wave in an ‘ideal’ estuary (expressed in \( h \) and \( u \)) may also be described as a ‘progressive wave’ (expressed in \( h_p \) and \( u \)) if we look at longitudinal compression only, as is the case in a prismatic channel where there is no lateral compression.
5.3. Apparent vs. Classical Reflection

The open boundary equation (1.1) has been derived in Sect. 2.1 for a ‘perfect channel’. However, if there is a closed boundary, then at this boundary the velocity will remain zero but the water level will change in time. Therefore neither invariant (see Eq. (1.7)) will be constant in time at this location. Hence, it can be concluded that Eq. (1.1) cannot apply to a standing wave in a finite, frictionless, prismatic, horizontal channel.

Equation (1.1) also appears to hold in simulations of converging channels with open boundaries, as shown in Sect. 3.3.3, even though in these estuaries a clear phase lag can be observed. However, it does not hold in such converging estuaries if there is a closed boundary.

Since it appears to be the ‘classical reflection’ which occurs at a closed boundary that causes Eq. (1.1) not to hold, it can be reasoned that if Eq. (1.1) does hold in converging estuaries with an open boundary, there is no such reflection. So the numerical results in Sect. 3.3.3 confirm that the phenomenon of ‘apparent reflection’ (by convergence, friction and bed slope) is fundamentally different from ‘classical reflection’, as already noted in the literature [6–8, 14, 16].
Figure 6.1: So, but what’s the application?

*My powers are ordinary.*

*Only my application brings me success.*

Isaac Newton
6.1. Reconstructing the Velocity

Through substitution of Eqs. (1.2) and (1.3) in Eq. (1.1), the spatial and temporal derivatives of the velocity can be removed, yielding a second order equation in $u$:

$$f_1 u^2 + f_2 u + f_3 = 0,$$

where

$$f_1 = \zeta_x (h_x - \beta h \pm f_W h),$$
$$f_2 = h_t h_x + h_t \zeta_x - h_t \beta h,$$
$$f_3 = h_t^2 - g h \zeta_x^2,$$
$$f_W = \frac{W}{u|u|}.$$  \hspace{1cm} (6.2)

If convergence, friction and bed slope are known, Eq. (6.1) can be used to compute the velocity $u$ directly out of observation of the water depth ($h$, $h_t$ and $h_x$). But there are some issues that have to be addressed.

1. Observations cannot be completely local since spatial derivatives are required.
2. The specific friction term $f_W$ has to be determined, which requires the local friction coefficient $K$ to be estimated, or calibrated as discussed in Sect. 6.2. Here it is assumed that in open channels the full friction term $W$ contains a quadratic velocity component, which is positive or negative depending on the flow direction.
3. Since this equation is of second order, it contains a part that can either be positive or negative, which changes within each simulation in a wave-like behavior.
Figure 6.2: The resulting velocity from Eq. (6.1) is compared with the numerical velocity in time series. The parameters used in the simulation are $\eta = 1.2 \text{ m}$, $h_0 = 28 \text{ m}$, $b = 280 \text{ km}$, $d = 1111 \text{ km}$, $K = 50 \text{ m}^1/\text{s}^{3/2}$.

Figure 6.2 illustrates the application of Eq. (6.1), in which the sign of the specific friction term $\pm f_V$ is taken from the numerical velocity and the sign of the quadratic root switches at the vertical asymptote. As can be observed, large spikes occur, which seem to introduce errors that dampen, but remain until the next spike.

Since numerical evaluations of the governing equations contain numerical errors, substitution may reduce or amplify the numerical error. Even though the proposed equation (6.1) is quite sensitive, in particular to small values of $f_V$, it is an interesting method to obtain the discharge flowing through the estuary from the easily observed water depth. In this sense, it functions as a stage-discharge method for the tidal zone.
6.2. Estimation of Friction

A second application comes from removal of the spatial derivatives of both the water depth and the velocity, again through substitution, in order to determine the friction term $W$. Although the width convergence $\beta$ and the bed slope $Z_x$ cannot be measured locally, it is a large advantage of this method that only local time series are required of the water depth $h$ and the velocity $u$. This can be done with

$$W = \frac{f_4 - f_5 f_6}{f_7}, \quad (6.3)$$

where

$$f_4 = g(\dot{h})^2 - h(\dot{u})^2,$$
$$f_5 = u (Z_x + \beta h),$$
$$f_6 = \left(\frac{u^2}{2} + g h\right)_t,$$
$$f_7 = (uh)_t. \quad (6.4)$$
Figure 6.3: The result of Eq. (6.3) is compared with three estimates of the friction term \( W \) from Eq. (3.8), in which the numerical velocity has been used. The parameters are from the same simulation as in Fig. 6.2.

A numerical test is shown in Fig. 6.3. The friction term \( W \) from Eq. (6.3) is compared to that obtained using Eq. (3.8) with three different values of the Strickler coefficient \( K \). As the friction relates to the square of the velocity, the signal also shows pronounced spikes. The best agreement is found for \( K = 50 \, m^{1/3} s^{-1} \), which is the value used in the numerical simulation. Therefore Eq. (6.3) may be suitable for local friction calibration.
Politics is for the present, but an equation is for eternity.

Albert Einstein
7.1. Summary

An exact ‘open boundary equation’ (1.1) has been derived for a progressive wave in a frictionless, prismatic, horizontal and infinite channel, under sub-critical flow, which is valid in the entire modeling domain. It was further investigated through numerical simulations if this equation had more general validity in estuaries that experience convergence, friction and bed slope.

Numerical simulations with a weakly reflective boundary yielded satisfactory results for all amplitude-to-depth ratios, in weakly and in strongly converging estuaries, with low and high friction and with bed slope, as long as Eq. (1.1) was evaluated at sufficient distance from the imposed boundary conditions.

The open boundary equation is an integral part of the problem formulation, representing the information of the landward open boundary. In this way, the open boundary equation is an additional third equation to the two balance equations (1.2) and (1.3), providing limits to the solution space.

The open boundary equation does not work if there is ‘classical reflection’ from a closed boundary. However, it does work if there is ‘apparent reflection’ (by convergence, friction and bed slope), which is fundamentally different from ‘classical reflection’.

The new equation can be used in practical applications, where it allows to remove two partial derivatives in the temporal and spatial variations of water elevation or velocity. In particular, removing the spatial derivatives of the velocity and water depth yields Eq. (3.5) for a fully absorbing, local boundary for the nonlinear Saint-Venant equations accounting for both friction and convergence. Alternatively, removing the velocity space and time derivative, Eq. (6.1) for the velocity \( u \) can be obtained from the water depth \( h \) and its derivatives in time \( h_t \) and space \( h_x \). Moreover, Eq. (6.3) is found with which friction can be calibrated locally from time series observations of both velocity and water depth.

Although there are substantial hints that Eq. (1.1) is indeed valid for any infinite channel, also with friction and convergence, this cannot formally be proven yet. Numerical simulations contain errors and can only examine an intrinsically limited region of the parameter space on a finite domain, so we recommend that further research is performed on this approach to look for formal proof.
7.2. Further Research
For further research, first funding will have to be acquired.
Since the open boundary equation can be considered as a key, several topics for follow up research can be done in random order. The research can be divided in theoretical research, where further computer model research is applied, and practical research, where the theory would be applied in nature.

For theoretical research, the following steps are proposed:

- Work on the seaward boundary condition. Prove that not every forcing (seaward) boundary condition can be posed on an estuary and show that only particular combinations of water level and velocity should be posed.
- Work on the landward boundary condition. Design a perfectly absorbing, local, open boundary.
- Apply additional terms in the momentum balance (e.g. wind), and see whether the same success can be obtained.
- Search for equivalent equations in 2-D (which requires strong computational power).
- If there is success in 2-D, search for equivalent equations in 3-D (which requires very strong computational power).

For practical research, the following steps are proposed:

- Measure the open boundary equation directly, somewhere in nature, which should be done before any other steps are taken.
- Elaborate on the tidal rating curve, see Sect 6.1.
- Improve the empirical friction formula with in-situ measurements, see Sect 6.2.
Whatever fights monsters should see to it that in the process he does not become a monster. And if you gaze long enough into an abyss, the abyss will gaze back into you.

Friedrich Nietzsche
A.1. General Description
I followed the courses of the master program of water management analogous to my own research. Therefore I did not work full time on the research, but only in free time when I was in the mood.

The main reference for the taken steps is the book ‘Salinity and Tides’ by Savenije [14]. What was used in this book is the Lagrangean reference frame (following particles), which is the standard in basic high school physics. In fluid dynamics the Eulerian frame (local) seems to be the standard, as it has some advantages for studying a (water) system. So I studied these two perspectives first.

What followed was a Wikipedia ‘Riemann Invariant’ study to obtain knowledge on the third important perspective in wave theory. This perspective diagonalizes the matrix which leads to invariants which move with the wave celerity, which is a crucial parameter for most of the current approximated wave solutions.

A.2. Additional thesis
The graduation work had to be cut in two. Clearly, it was a single process. The following work is described in the additional thesis.

- Exact solution of Savenije’s mass balance for estuaries,
- Lagrangean Analysis,
- Lagrangean Curve Fitting,
- Draft version paper “Eulerian versus Lagrangean framework”,
- Wave Character with Particle Tracking,
- Phase Analysis,
- Draft version paper “New equation, in different perspectives”.

The main result, the open boundary equation, was further explored in this master thesis.

A.3. Master thesis
The result of this work is the discovery of a special relation in the 1-Dimensional Saint-Venant equations. They are among the most used equations in the world of water management and are the basis for more complex equations for water movement.

After finding this interesting relationship about a year of work followed. The biggest part of the work was setting up the numerical scenarios and writing scripts to interpret the results in order to test the hypothesis. Only some of the MATLAB scripts are in this report, which in fact is a summary.
Computers are magnificent tools for the realization of our dreams, but no machine can replace the human spark of spirit, compassion, love, and understanding.

Louis V. Gerstner, Jr.
**B.1. Numerical Model Toffolon**

**B.1.1. Governing Equations**

The used numerical model evaluates the following equations:

\[ h_t + uh_x + hu_x - \beta uh = 0 , \]
\[ u_t + uu_x + g\zeta_x + W = 0 , \]
\[ h = \zeta - Z , \quad \beta = -\frac{B_x}{B} \]

where:

- \( x \) is the longitudinal coordinate (positive landward),
- \( t \) is the time,
- \( g \) is the acceleration due to gravity,
- \( u(x, t) \) is the cross-sectional average flow velocity,
- \( h(x, t) \) is the water depth,
- \( \zeta(x, t) \) is the water level,
- \( Z(x) \) is the bed level,
- \( B(x) \) is the width and
- \( W(x, t) \) is the frictional term.

**B.1.2. Numerics**

- The numerical model by Toffolon et al. [17] has been written in fortran.
- The model uses Time integration by McCormack explicit, whereby it applies a total variation dissipation-filter (TVD).
- The spatial resolution was chosen as \( dx = L/1000 \), which means that over 500 gridpoints where used per wavelength, since \( L = 2c_0T \), and the depth only decreased in landward direction.
- The time step was determined by using a maximum Courant wave number.

**B.1.3. Acknowledgments**

First \( \pm 30 \) simulations were provided by Cai Huayang. Marco Toffolon adjusted the code to do multiple scenarios, so that 10'000 scenarios could be evaluated at once.
B.2. MATLAB

For post-processing, MATLAB has been used. Over 100 (draft)-codes have been used in this work. The following codes have been used for the main, final analysis.

B.2.1. analysisDirk

- plot timeseries

% - analysis of the terms of the equations
% - other analyses by Dirk

%close all;
format bank

if verb==0
    display(’Verbosity is zero (no main variables)’)
    return
end

%choice parameters
keq=1; %1=eq.cross, 2=eq.sqxcf, 3=mass, 4=momentum, 5=ufromh, 6=friction
correplot=1; %1=correlation plot
correlpause=0; %pause for correl plot
scatterplot=1; %1=scatter plot
overview=0; %overview simulations
tseries=0; %1=timeseries analysis
locfun=0.4; %rand(n,1); %location of timeseries (x/L)
plotsign=0; %count analysis for ufromh

%start parameters
if(correplot==1)
    figd1= figure(’Position’,([1 1 scrsz(3) scrsz(4)]));
end
if(scatterplot==1)
    figd2=figure;
end
fmin=0; %for red line
fmax=0; %for red line
hs=es*0; %preallocation for storage depth
sign=zeros(n,m(1));

legCorr=[];

col=[0 0 1; %blue
     1 1 0; %yellow
     0 1 0; %green]
1 0 0; %red
0 1 1; %cyan
1 0 1]; %magenta

f1mat=0*es;
f1amat=f1mat;
f1bmat=f1mat;
f1cmat=f1mat;
f2mat=f1mat;
f2amat=f1mat;
f2bmat=f1mat;

% calc
i2=0;
for i=1:n
    %select the proper range of data for the simulation
    i1=i2+1; i2=i2+ntsteps(i);
    t=ts(i1:i2)*T(i); %time

    %main variables
    e=es(i1:i2,:); %water level
    h=bsxfun(@minus,e,bed(i,:)); %depth
    hs(i1:i2,:)=h; %storage depth
    u=us(i1:i2,:); %velocity
    R1=u+2*sqrt(g*h);
    R2=u-2*sqrt(g*h);

    %gradients
    dt=gradient(t);
    [havg,Dt]=meshgrid(-bed(i,:),dt);
    [ex,et]=gradient(e); %water level
    ex=ex/dx(i);
    et=et./Dt;
    [hx,ht]=gradient(h); %depth
    hx=hx/dx(i);
    ht=ht./Dt;
    [havgx,havgt]=gradient(havg); %average depth
    havgx=havgx/dx(i);
    havgt=havgt./Dt;
    I=-havgx; %bottom slope
    [ux,ut]=gradient(u); %velocity
    ux=ux/dx(i);
    ut=ut./Dt;
beta=-gradient(width(i,:))/dx(i)./width(i,:);
[Beta,bla]=meshgrid(beta,linspace(0,0,512));

%check fortran/matlab
if(numterms==1)
    ht_n=T_dHdt(i1:i2,:);
    uhx_n=T_UdYdx(i1:i2,:);
    hux_n=T_YdUdx(i1:i2,:);
    uyb_n=T_UYdBdx(i1:i2,:);
    subplot(2,2,1)
    plot(x,ht(200,:));
    hold on
    plot(x,ht_n(200,:),'r');
    subplot(2,2,2)
    plot(x,u(200,:).*hx(200,:));
    hold on
    plot(x,uhx_n(200,:),'r');
    subplot(2,2,3)
    plot(x,h(200,:).*ux(200,:));
    hold on
    plot(x,hux_n(200,:),'r');
    subplot(2,2,4)
    plot(x,-u(200,:).*h(200,:)*ib(i));
    hold on
    plot(x,uyb_n(200,:),'r');
    return
end

%calc
switch keq
    case 1 %eq. 13
        % LHS and RHS (scaled)
        f1=et.*ux /scale(i);
        f2=ut.*ex /scale(i);

        % storage of results
        if(tseries==1)
            f1mat(i1:i2,:)=f1;f2mat(i1:i2,:)=f2;
            havgs(i1:i2,:)=havg;
        end
    case 2 %eq. 21
        % 4 terms
        f1a=ux.^2.*h
            /scale(i);
flb = u.*(hx-ex) .*ux /scale(i);
flc = -u.*Beta.*h .*ux /scale(i);
f2a = g*ex.^2 /scale(i);
W = g*u.*abs(u)./(K(i)^2*h.^(4/3));
f2b = W.*ex /scale(i);

% LHS and RHS
f1 = (f1a+f1b+f1c);
f2 = (f2a+f2b);

% storage of results
if(tseries==1)
f1mat(il:i2,:) = f1;
f2mat(il:i2,:) = f2;
end

% terms mass
f1 = ht + u.*hx ;
f2 = -h.*ux + u.*h.*Beta ;

% storage of results
if(tseries==1)
f1mat(il:i2,:) = f1;
f2mat(il:i2,:) = f2;
havgs(il:i2,:) = havg;
end

% terms momentum
W = g*u.*abs(u)./(K(i)^2*h.^(4/3));
fla = ut;
flb = u.*ux;
flc = g*hx;
fld = W;

f1 = ut + u.*ux + g*hx - g*havgx;
f2 = -W;

% storage of results
if(tseries==1)
f1mat(il:i2,:) = f1;
f2mat(il:i2,:) = f2;
flamat(il:i2,:) = fla;
flbmat(il:i2,:) = flb;
flcmat(il:i2,:) = flc;
fldmat(il:i2,:) = fld;
havgs(il:i2,:) = havg;
end

% u from h
f1a = u;
f1 = f1a;

\%W = g/K(i)^2 * u.*abs(u)./h.^(4/3);
fW = g/K(i)^2 ./h.^(4/3);

formAPos = ex .* (hx - Beta.*h) - h.*fW.*ex;
formANeg = ex .* (hx - Beta.*h) - h.*(-fW).*ex;
signPos = (u>0)*1; \% decide if friction is positive or negative
signNeg = 1-signPos;

formA = signPos.*formAPos + signNeg.*formANeg;
formB = ht .* (ex + hx - Beta.*h);
formC = ht.^2 - g*h.*ex.^2;
formDet = formB.^2 - 4*formA.*formC;

f2a = -formB ./(2*formA);
f2b = real(sqrt(formB.^2 - 4*formA.*formC)) ./(2*formA);

f2plus = f2a + f2b;
f2min = f2a - f2b;
signPlus = (abs(f2plus-u) < abs(f2min-u))*1; \% decide to take the plus or minus in the ABC formula, which works best
signMin = 1-signPlus;

f2 = signPlus.*f2plus + signMin.*f2min; \% final u from h

\% storage of results
if(tseries==1)
    f1mat(i1:i2,:) = f1; f2mat(i1:i2,:) = f2;
    f1amat(i1:i2,:) = f1a; f2amat(i1:i2,:) = f2a; f2bmat(i1:i2,:) = f2b;
    f2plusmat(i1:i2,:) = f2plus; f2minmat(i1:i2,:) = f2min;
    formAmat(i1:i2,:) = formA; formBmat(i1:i2,:) = formB; formCmat(i1:i2,:) = formC;
    formDetmat(i1:i2,:) = formDet;
    signPlusmat(i1:i2,:) = signPlus;
    havgs(i1:i2,:) = havg;
end

\% friction
diffk = 10;
W1 = g/(K(i)-diffk)^2 * u.*abs(u)./h.^(4/3);
W2 = g/K(i)^2 * u.*abs(u)./h.^(4/3);
W3 = g/(K(i)+diffk)^2 * u.*abs(u)./h.^(4/3);
fla = W1;
\[ E = \frac{1}{2} u^2 + g h \] 
\[ \{Ex, Et\} = \text{gradient}(E) \]
\[ Ex = \text{Ex} / \text{dx}(i) \]
\[ Et = \text{Et} / \text{Dt} \]
\[ q = u^* h \]
\[ \{qx, qt\} = \text{gradient}(q) \]
\[ qx = \text{qx} / \text{dx}(i) \]
\[ qt = \text{qt} / \text{Dt} \]
\[ f4 = g \times h^2 - h \times u^2 \]
\[ f5 = u \times (h_\text{avg}x - \text{Beta} \times h) \]
\[ f6 = \text{Et}; \quad u \times u + g \times h \]
\[ f7 = \text{qt}; \quad u \times h + h \times u \]
\[ f2a = f4 / f7 \]
\[ f2b = f5 \times f6 / f7 \]
\[ f1 = f1b \]
\[ f2 = (f4 + f5 \times f6) / f7 \]

% storage of results
if(tseries==1)
    \[ f1mat(i1:i2,:) = f1; f2mat(i1:i2,:) = f2; \]
    \[ flmat(i1:i2,:) = f1a; f1bmat(i1:i2,:) = f1b; f1cmat(i1:i2,:) = f1c; \]
    \[ f2amat(i1:i2,:) = f2a; f2bmat(i1:i2,:) = f2b; \]
    \[ havg(i1:i2,:) = \text{havg}; \]
end

% correlation plot
for j=1:m(i)
    correl(i,j) = \text{corr}(f1(:,j), f2(:,j));
end

if(correlplot==1)
    if(1) % F_status(i)==0
        legCorr = [legCorr; '\eta=' num2str(Amp(i)) 'm'];
        figure(figd1)
if(correlpause==1)
    plot(correl(i,:),'color',rand(3,1))
    title(['Scenario ' num2str(i) '.'])
    ylim([.5 1])
    pause
else
    hold on
    if i>6
        plot(correl(i,:),'color',rand(3,1))
    else
        plot(correl(i,:),'color',col(i,:)) %x(i,:),
    end
end
end

% scatter plot
selt=ntsteps(i)/16:ntsteps(i)/16:ntsteps(i); %selected points to plot
dm=ceil((m(i)-1)/10);

selx=3*dm:dm/2:7*dm;
%selxInp=3*dm:dm/2:7*dm;
%selx=selxInp(find(correl(i,selxInp)>.99)); %cthr=.99

if(scatterplot==1)
    f1x=f1(selt,selx);
    f2x=f2(selt,selx);
    f1scat=reshape(f1x,size(f1x,1)*size(f1x,2),1);
    f2scat=reshape(f2x,size(f2x,1)*size(f2x,2),1);
    figure(figd2)
    hold on
    scatter(abs(f1scat),abs(f2scat),4,'filled')
    fmin=min([min(f1scat),min(f2scat),fmin]);
    fmax=max([max(f1scat),max(f2scat),fmax]);
end

% overview for interpretation: water level, velocity and Riemann invariants
if(overview==1)
    figure('Position',[1 1 scrsz(3) scrsz(4)])
    subplot(221)
    mesh(h)
    title(['h'])
subplot(222)
mesh(u)
title('u')
subplot(223)
contourf(R1)
title('R1')
subplot(224)
contourf(R2)
title('R2')
end
end

% layout correlation and parity figures
if(correlplot==1)
  figure(figd1)
  %title('Pearson correlation plot')
  %legCorr=num2str((1:i)');
  legend(legCorr)
  xlabel('x [m]')
  ylabel('ho')
  %axis([0 100000 .5 1])
  %set(gca,'XTick',[0,5,10]*10^4,'YTick',[.5,.6,.7,.8,.9,1]);
end
if(scatterplot==1)
  figure(figd2)
  %title('Parity plot')
  %legend(legPar)
  xlabel('|\zeta_t u_x|')
  ylabel('|u_t \zeta_x|')
  hold on
  plot([10^-2 10^6],[10^-2 10^6],'-r','linewidth',1)
  axis([10^-2 10^6 10^-2 10^6 ])
  set(gca,'XTick',[.01 1 100 10000 1000000], 'YTick', [.01 1 100 10000 1000000])
end

Undefined function or variable 'verb'.

Error in picParity (line 7)
if verb==0

plot timeseries

if(tseries==1)
for i=[25 81 95 99] %:n
    ind=(1+(i-1)*ntsteps(i):i*ntsteps(i)); % time index
    loc=ceil(locfun.*L./dx);
    loc=loc(i);
    col1= [1 0 0]; % rand(1,3)
    col2= [0 1 0]; % rand(1,3)
    col3= [0 0 1]; % rand(1,3)
    col4= [0 1 1]; % rand(1,3)
    col5= [0 0 0]; % rand(1,3)
    zoom=10;

    ymin=min( f1mat(ind,loc) );
    ymax=max( f1mat(ind,loc) );

    % visual check LHS vs RHS
    figure('Position',[1 1 scrsz(3) scrsz(4)])

    switch keq
    case 2
        subplot(3,2,1)
        hold on
        plot(ts(ind),f1amat(ind,loc),'color',col1)
        plot(ts(ind),f2amat(ind,loc),'color',col2)
        legend('f1a','f2a')
        xlabel('t/T [-]')
        subplot(3,2,3)
        hold on
        plot(ts(ind),f1bmat(ind,loc),'color',col3)
        plot(ts(ind),f1cmat(ind,loc),'color',col4)
        plot(ts(ind),f2bmat(ind,loc),'color',col5)
        legend('f1b','f1c','f2b')
        xlabel('t/T [-]')
        ymin=1.5*min(min(f1mat(ind,loc)));
        ymax=1.5*max(max(f1mat(ind,loc)));
        ylim([ymin ymax])
    case 4
        subplot(3,2,[1 3])
        hold on
        plot(ts(ind),f1amat(ind,loc),'color',col1)
        plot(ts(ind),f1bmat(ind,loc),'color',col2)
        plot(ts(ind),f1cmat(ind,loc),'color',col3)
        plot(ts(ind),f1dmat(ind,loc),'color',col4)
legend('u_t', 'u u_x', 'g h_x', 'W')
xlabel('t/T [-]')

ymin = 2*min(min(fldmat(ind, loc))); ymax = 2*max(max(fldmat(ind, loc))); ylim([ymin ymax])

case 5
subplot(3,2,1)
hold on
plot(ts(ind), f2plusmat(ind, loc), 'color', col1)
plot(ts(ind), f2minmat(ind, loc), 'color', col2)
plot(ts(ind), us(ind, loc), 'color', col5)
legend('plus', 'min', 'u')
axis([0 1 min(min(us(ind, loc))) max(max(us(ind, loc)))])
xlabel('t/T [-]')
title(['\eta_0=' num2str(Amp(i)) ', h_0=' num2str(h0(i)) ', K=' num2str(K(i))])

ymin = 2*min(min(us(ind, loc))); ymax = 2*max(max(us(ind, loc))); ylim([ymin ymax])

subplot(3,2,3)
hold on
%plot(ts(ind), formDetmat(ind, loc), 'color', col3)
plot(ts(ind), formAmat(ind, loc), 'color', col3)
plot(ts(ind), formBmat(ind, loc), 'color', col4)
plot(ts(ind), formCmat(ind, loc), 'color', col5)
legend('A', 'B', 'C')
xlabel('t/T [-]')

case 6
subplot(3,2,[2 4])
hold on
plot(ts(ind), f2mat(ind, loc), '-k')
plot(ts(ind), flamat(ind, loc), '-r')
plot(ts(ind), flbmat(ind, loc), '-g')
plot(ts(ind), flcmat(ind, loc), '-b')
legend('W Eq.34', [W (K=' num2str(K(i)-diffk) ')'], [W (K=' num2str(K(i)+diffk) ')'], [W (K=' num2str(K(i)) ')'])
%legend('f4/f7', 'f5*f6/f7')
xlabel('t/T [-]')
title(['\eta_0=' num2str(Amp(i)) ', h_0=' num2str(h0(i)) ', b=' num2str(round(b(i))) ', d=' num2str(round(d(i))) ', K=' num2str(round(K(i)))'])

ymin = 3*min(flamat(ind, loc)); ymax = 3*max(flamat(ind, loc));
ylim([ymin ymax])
continue
end

subplot(3,2,[2 4])
plot(ts(ind),f1mat(ind,loc),'-g',ts(ind),f2mat(ind,loc),'-r')
legend('f1','f2')
xlabel('t/T [-]')
title([Rho= num2str(correl(i,loc)) [-], Scenario num2str(isim(n1-1+i)) ]) %
ylim([ymin ymax])

subplot(3,2,5)
hold on
plot(ts(ind),es(ind,loc),'-k')
plot(ts(ind),us(ind,loc),'-g')
legend('\zeta','u')
xlabel('t/T [-]')
title([Location x/L = num2str(ceil(100*loc/m(i))/100) [-], x = num2str(ceil(loc/m(i)*L(i)/1000)) km])

subplot(3,2,6)
plot(x(i,:),-exp(-x(i,:)/b(i)),'-k',x(i,:),-havgs(i*ntsteps(i,:),)/havgs(i*ntsteps(i,:),1))
title([Channel with b= num2str(b(i)) , d= num2str(d(i)) , K= num2str(round(K(i)))])
legend('B/B0','h_{avg}/h_0')
xlabel('x [m]')
ylim([-1 0])

if (plotsign==1)
    figure('Position',[1 1 scrsz(3) scrsz(4)])
    subplot(2,2,1)
    contourf(es(ind,:))
colorbar
title({'\zeta; h_0= num2str(h0(i)) ', \eta= num2str(eps(i)*h0(i))})
    subplot(2,2,2)
    contourf(us(ind,:))
colorbar
title('u')
    title([Scenario ' num2str(i) '])
    subplot(2,2,3)
    contourf(signPlusmat(ind,:))
colorbar
title('sign in ABC formula (1=plus, 0=minus)')
    subplot(2,2,4)
    hold on
    contourf(log(f2bmat(ind,:)))
colorbar

title('Log sqrt(B^2-4AC)')
%eval(['print -dpng ' dirModel 'countUfromhPlusMinScen' num2str(i) '.png'])
end
end

if(mov==1)
% movie for visual check LHS vs RHS
speed=10;
figure('Position', [1 1 scrsz(3) scrsz(4)]);
for i=1:n
    ind=(i-1)*ntsteps(i)+1:speed:i*ntsteps(i)-1;
ymin=min(min(f1mat(ind,:)),min(f2mat(ind,:)))
ymax=max(max(f1mat(ind,:)),max(f2mat(ind,:)))
    for j=ind
        subplot(2,1,1)
        plot(x(i,:),us(j,:),'-g',x(i,:),es(j,:),'-k')
        subplot(2,1,2)
        plot(x(i,:),f1mat(j,:),'-g',x(i,:),f2mat(j,:),'--r')
        % ylim([ymin ymax])
        title(['h_t*u_x (green) versus u_t*h_x (red) in scenario ' num2str(isim(n1-1+i)])
        pause(1/speed)
    end
end
end
### B.2.2. `picLagrCorrWidth`

```matlab
if (n>1)
    display('n>1! Only for one simulation!')
    return
end

% time
t=t0;
t0=0;
tdim=t*T;
dt_dim=gradient(tdim);
dt=diff(t);

% space
x_p=zeros(1,length(t));
x_p(1)=L*0.2;  % starting point
x0=x_p(1);

% variables
u=us;
e=es;
h=bsxfun(@minus,es,bed);
[B,bla]=meshgrid(width,t);

uL=linspace(0,0,ntsteps);
hL=linspace(0,0,ntsteps);
eL=linspace(0,0,ntsteps);
BL=linspace(0,0,ntsteps);

%-- particle tracking
for i=1:ntsteps-1
    uL(i)=interp2(x,t,u,x_p(i),t(i));
    k_1 = uL(i);
    k_2 = interp2(x,t,u,x_p(i)+0.5*k_1*dt_dim(i),t(i)+0.5*dt(i));
    k_3 = interp2(x,t,u,x_p(i)+0.5*k_2*dt_dim(i),t(i)+0.5*dt(i));
    k_4 = interp2(x,t,u,x_p(i)+k_3*dt_dim(i),t(i)+dt(i));
    x_p(i+1) = x_p(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*dt_dim(i);
end
i=length(t);
uL(i)=interp2(x,t,u,x_p(i),t(i));
for i=1:length(t)
    hL(i)=interp2(x,t,h,x_p(i),t(i));
eL(i)=interp2(x,t,e,x_p(i),t(i));
    BL(i)=interp2(x,t,B,x_p(i),t(i));
end
```
S=x_p-x_p(1);
hp=hL.*BL/BL(1);
h0L=hL-eL;

%% plot
close all;
figure
subplot(211)
hold on
plot(t,uL,'-g')
plot(t,hL-h0L,'-k')
plot(t,hp-h0L,'-r')
legend('u [m/s]','h - h_0 [m]','h_p - h_0 [m]')
xlabel('t/T [-]')
set(gca,'XTick',[0 .5 1])

subplot(212)
hold on
plot(S/1000,uL,'-g')
plot(S/1000,hL-h0L,'-k')
plot(S/1000,hp-h0L,'-r')
xlabel('S [km]')
xlim([-7.5 7.5])
set(gca,'XTick',[ -5 0 5 ])
Additional Simulations

C.1. Time series

Numerical results are presented in the series of Figures C.1-C.6. The organization of the plots is the same in all these figures. Water depth $h$ and velocity $u$ in the whole domain are displayed in the upper and lower plots, respectively, of the left column. In the upper right plot, the time series of $h$ and $u$ are shown. In the lower right plot, the time series of the left and the right hand sides of equation (1.1) are plotted. As specified in the figures, time series are taken at a reasonable distance from the boundaries, to avoid the influence of the seaward boundary condition.

Scenarios 1-4 are performed by SWASH [18]. In these simulations the channels are horizontal and prismatic. Moreover, here Chezy friction is used, which relates to Strickler friction by $C = Kh^{1/6}$.

Scenarios 5-6 are performed by the model of Toffolon [17]. In these simulations the channels are horizontal and exponentially converging, see Sect. 3.3.1. Moreover, here Strickler friction is used.
Figure C.1: Scenario 1: $\bar{h} = 10$ m, $\eta = 0.001$ m, $c = 50$ m$^{1/2}$ s$^{-1}$.

Figure C.2: Scenario 2: $\bar{h} = 10$ m, $\eta = 0.2$ m, $c = 20$ m$^{1/2}$ s$^{-1}$.
Figure C.3: Scenario 3: $\bar{h}=10$ m, $\eta=3$ m, $c=100$ m$^{1/2}$s$^{-1}$.

Figure C.4: Scenario 4: $\bar{h}=10$ m, $\eta=9$ m, $c=50$ m$^{1/2}$s$^{-1}$.
Figure C.5: Scenario 5: $\bar{n}=20$ m, $\eta=10$ m, $\kappa=26$ m$^{1/3}$ s$^{-1}$, $b=40$ km.

Figure C.6: Scenario 6: $\bar{n}=12$ m, $\eta=1.2$ m, $\kappa=43$ m$^{1/3}$ s$^{-1}$, $b=70$ km.
Only the very weak-minded refuse to be influenced by literature and poetry.

Cassandra Clare

References


