University of Bath

Department of Mathematical Sciences

MSc in Modern Applications of Mathematics

Academic Year 2002-2003

MSc Project

Transcritical flow modelling with the Box Scheme

Melina Freitag

Supervisor: Prof. K.W. Morton

September 18, 2003

Declaration

I declare that this document and the accompanying code has been composed by myself, and describes my own work, unless otherwise acknowledged in the text. It has not been accepted in any previous application for a degree. All sources of information have been specifically indicated.

Acknowledgements

This project would not have been possible without the guide and support of Professor K.W. Morton. I would like to thank him for his continued help and advice throughout this work.

To Daniel

Abstract

We investigate the Preissmann Box Scheme which is the standard numerical scheme used by hydraulic engineers to model open channel flows or surcharged flows in pipes.

We apply the Scheme to both subcritical and transcritical flow and analyse its behaviour in both cases. Since the Box Scheme breaks down in the second case we present a method, which overcomes this problem.

By adopting the work which has been done for steady transonic flows and applying it to unsteady open channel flows we develop a *modified Box Scheme* which works efficiently for transcritical flows. Numerical examples supporting our results are presented.

Analysis of the solution of nonlinear systems is used together with computational testing of new algorithms.

Contents

List of Figures

1 Introduction

The accurate computer simulation of river and pipe flows is of great importance in the design of urban drainage networks. The St Venant equations are widely used by Hydraulic Engineers as an accurate model for one-dimensional open channel flow and surcharged flow in pipes. However, the analytic solution to these nonlinear equations is only known for a limited number of special cases. Therefore we need a reliable and accurate numerical solver.

Finite difference schemes, such as the Preissmann Box Scheme [27], are the standard schemes used in the solution of the St Venant equations. The Box Scheme, as an implicit method, is unconditionally stable and extremely robust as it allows one to choose the time step on the basis of accuracy rather than stability. It leads to a nonlinear system of equations, which may be solved iteratively, using Newton's Method. Using its block-tridiagonal structure, the resulting system can be solved relatively cheaply with the Thomas Algorithm. [24].

Figure 1: *Supercritical flow with hydraulic jumps under a bridge*

The Box Scheme can be applied to both purely subcritical and purely supercritical flow. However, the occurrence of transcritical flow (i.e. the existence of both subcritical and supercritical flow regions in the considered domain) leads to problems with the original implementation of the Box Scheme and the method breaks down. We describe these problems and the reasons for their occurrence.

Transcritical flow occurs frequently in channels with rapidly changing channel width or on steep slopes, for example in mountainous areas. Steep slopes sometimes lead to the formation of shocks, that is stationary or moving hydraulic jumps. Another occurrence of transcritical flow is for dam-break waves.

There have been many attempts to overcome this limitation of the Box Scheme. One approach is to change the underlying system of differential equations, such that the flow remains subcritical [9, 11]. This sometimes gives reasonable results, but is physically incorrect.

The correct approach is to keep the differential equations unmodified and switch between the subcritical and supercritical implementation of the Box Scheme. Since the mathematical situation is similar to that for compressible gas flow in a transonic duct, we can use the concept of residual distribution, which was introduced in Morton et al. [25] and successfully applied to the steady-state Euler equations. We want to adapt this technique to the unsteady St Venant equations, and show that we can accurately model time dependent transcritical flow. Several test problems will be carried out and we will see in the numerical results, that even transcritical problems involving shocks are modelled accurately. All computations are carried out in MATLAB [8]. The description of all code developed and used for this report may be found in the appendices.

2 The Saint Venant Equations

In this section we will give a derivation of the Saint Venant equations and apply them to a channel with trapezoidal cross-section. We will use these equations throughout this report as model equations for open channel flow.

2.1 Assumptions and Derivation

The St Venant equations are generally used to provide an accurate model of one dimensional open channel flow. They are essentially obtained from the principles of mass and momentum conservation. In order to derive them, certain assumptions have to be made:

- the flow is essentially 1-D,
- the water in the channel is an incompressible ideal fluid and has constant density,
- the pressure is hydrostatic,
- all forces are due to gravity and friction, i.e. forces due to wind stress, Coriolis force, atmospheric pressure etc. are negligible,
- the channel bed does not change in time,
- the volumetric inflow due to rain, tributaries and evaporation is negligible.

2.1.1 Mass Conservation

Using the assumptions of uniform density and incompressibility of water in a river, the continuity equation of the Navier-Stokes equations (see [26]) reduces to

$$\nabla \cdot \mathbf{u} = \nabla \cdot (u, v, w) = 0, \tag{2.1}$$

where $\mathbf{u}$ is the velocity vector field. Without any assumptions on the x, y-variation of the surface we now average over the depth $h - b$, by integrating over z, see Figure

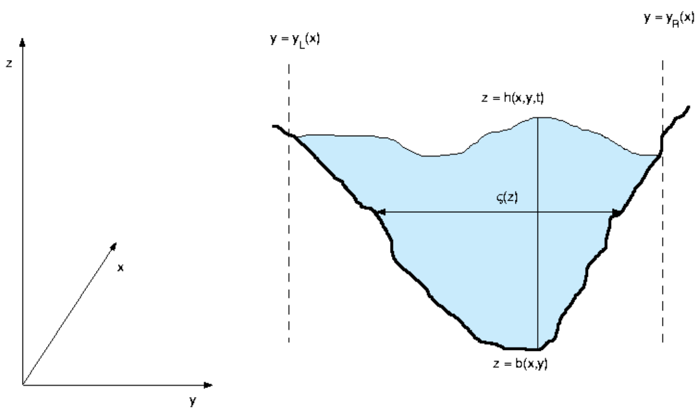

Figure 2: *Channel cross-section*

2. Furthermore we assume free surface conditions,

$$\frac{D}{Dt}(h(x,y,t) - z) = 0, \tag{2.2}$$

i.e. particles on the surface cannot leave the surface; and similarly

$$\frac{D}{Dt}(b(x,y) - z) = 0, \tag{2.3}$$

for the channel bed. Those equations simplify to

$$(\partial_t + u\partial_x + v\partial_y)\,|_{z=h} h - w|_{z=h} = 0, \tag{2.4}$$
$$(u\partial_x + v\partial_y)\,|_{z=b} b - w|_{z=b} = 0. \tag{2.5}$$

We define the vector $\mathbf{q}$ to be the integral of (u, v) over the depth, i.e.

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \int_{\eta=b}^{\eta=h} \begin{bmatrix} u \\ v \end{bmatrix} d\eta. \tag{2.6}$$

Using this equation and (2.1), (2.4) and (2.5) we get

$$\partial_t h + \nabla \cdot \mathbf{q} = 0, \tag{2.7}$$

the flow equation in terms of the height and the velocities. In order to complete the cross-sectional averaging, we assume the flow is parallel to the boundary and define the discharge

$$Q(x,t) = \int_{y_L(x)}^{y_R(x)} q_1 dy, \tag{2.8}$$

and the wetted cross-sectional area

$$A(x,t) = \int_{y_L(x)}^{y_R(x)} (h-b) dy, \tag{2.9}$$

so that we can integrate (2.7) to get

$$A_t + Q_x = 0. \tag{2.10}$$

Another way to derive this equation is to consider the change of mass in a cross-section. The principle of mass conservation states that the total change in mass is balanced by the flow through the boundary, with no flow through the bed or the surface. These assumptions lead to the integral form of the mass conservation, which was derived in [9],

$$\int_x^{x+\Delta x} [A]_t^{t+\Delta t} dx + \int_t^{t+\Delta t} [Q]_x^{x+\Delta x} dt = 0. \tag{2.11}$$

2.1.2 Momentum Conservation

Now we can apply the same argument as in the previous section to the x-momentum. From the basic Navier-Stokes equations (see [26]) we have

$$\mathbf{u}_t + \nabla \cdot (\mathbf{u}\mathbf{u} + pI) = F, \tag{2.12}$$

where $\mathbf{u} = (u, v, w)$ is the velocity vector field, p is the pressure and F represents all external forces. We need to make assumptions about the normal momentum flux on the boundary, the momentum, gravity and bed friction.

Again, by integrating over the cross-section in x-direction, the first term in (2.12) is given by

$$\int_x^{x+\Delta x} [Q]_t^{t+\Delta t} dx. \tag{2.13}$$

Assuming hydrostatic pressure, i.e. $p = p_{at} + g(h - z)$, the second term in (2.12) becomes

$$\int_{t}^{t+\Delta t}\left[\frac{\beta Q^2}{A}+gI_1\right]_{x}^{x+\Delta x}, \tag{2.14}$$

where $\beta \geq 1$ depends on the vertical variation of the velocities and

$$I_1(x,t)=\int_{b}^{h}\int_{y_L}^{y_R}(h-z)dydz \tag{2.15}$$

is the cross-sectional moment integral. The external forces give terms from the bed friction but also from $g(h-z)$, arising from the change in bed cross-section with x, and obtained from integrating $g(h-z)$ over the bed surface area projected into the x-direction. Hence, the external forces may be written as

$$\int_{t}^{t+\Delta t}\int_{x}^{x+\Delta x} g(A(S_0-S_f)+I_2)dxdt, \tag{2.16}$$

where S_0 is the bed slope, S_f is the frictional slope from the fricitional forces and

$$I_2(x,t)=\int_{b}^{h}\int_{y_L}^{y_R}(h-z)\sigma_x dydz \tag{2.17}$$

is the pressure force acting on the channel bed. The function σ_x represents the change in channel width in the x-direction. Hence, the momentum equation (2.12) becomes

$$\int_{x}^{x+\Delta x}[Q]_t^{t+\Delta t}dx+\int_{t}^{t+\Delta t}\left[\frac{\beta Q^2}{A}+gI_1\right]_{x}^{x+\Delta x}=\int_{t}^{t+\Delta t}\int_{x}^{x+\Delta x} g(A(S_0-S_f)+I_2)dxdt. \tag{2.18}$$

These equations are often derived in terms of the river width at each level rather than its depth at each y-position. In order to see the relationship between those two approaches, we refer to Figure 2 and consider first the description of the bed shape at each value of x.

In the (y,z)-plane the bed shape is given either by $z=b(y)$ or equivalently by $\Delta y = \varsigma(z)$, where $\varsigma(z)$ is the channel width at height z. Then we can introduce the bed level $\xi(x)$ and replace z by $\eta := z-\xi(x)$, a translation in the z-direction. Hence, the channel width may be written as $\sigma(x,\eta)=\sigma(\eta)-\varsigma(z)$. Then, from (2.9), we can define the river depth $D(x,t)$ by

$$\int_0^D \sigma(x,\eta)d\eta = A(x,t); \tag{2.19}$$

so it is really an average depth over the width of the channel, i.e, the depth does not depend on the y-variation of the water surface. As a result we can write

$$\begin{aligned} I_1(x,t) &= \int_0^{D(x,t)} (D(x,t)-\eta)\sigma(x,\eta)d\eta, && (2.20)\\ I_2(x,t) &= \int_0^{D(x,t)} (D(x,t)-\eta)\sigma_x(x,\eta)d\eta. && (2.21)\end{aligned}$$

Where required we define the river height as $h(x,t) = D(x,t) + \xi(x)$; note that the height of a river is more easily observed than its depth.
A more detailed derivation of the St Venant equations may be found in [29].

2.2 A General Differential Form

Using the derivation of the previous section a general differential form of the St Venant equations for unsteady flow is given by

$$\begin{aligned} \frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} &= 0, && (2.22)\\ \frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\beta Q^2}{A} + gI_1\right) &= gI_2 + gA(S_0 - S_f), && (2.23)\end{aligned}$$

provided the wetted cross-sectional area $A(x,t)$ and the discharge $Q(x,t)$ are sufficiently smooth. The distance along the channel is given by x, t is the time, $S_0(x)$ is the bed slope, $S_f(x,A,Q)$ is the frictional slope, g is the gravitational acceleration and β is the momentum coefficient. For an ideal fluid, with no boundary layers, we usually have $\beta = 1$. Furthermore, the bed slope S_0 is given by

$$S_0(x) = -\frac{d\xi}{dx}, \tag{2.24}$$

where $\xi(x)$ is the bed level, the elevation of the bed above some horizontal datum. $I_1(x,t)$ and $I_2(x,t)$ are given by (2.20) and (2.21). The average fluid velocity for each channel cross-section is given by $v(x,t) = \dfrac{Q(x,t)}{A(x,t)}$. Finally, for the frictional slope S_f

we use Manning's formula , stated in [9] with

$$S_f = \frac{Q|Q|}{K^2}, \tag{2.25}$$

where the conveyance $K(x,h)$ is given by

$$K = \frac{A^{5/3}}{nP^{2/3}}. \tag{2.26}$$

$P(x,h)$ is the wetted perimeter and n is Manning's coefficient, which measures the roughness of the channel. In the following section we describe how the general St Venant equations (2.22) and (2.23) can be written in the case of trapezoidal cross-sections.

2.3 Channel with Trapezoidal Cross-section

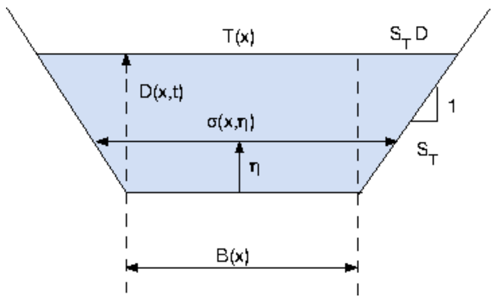

Figure 3: *Trapezoidal channel cross-section*

If the channel cross-section is trapezoidal, then the width of the channel can be described by

$$\sigma(x,\eta) = B(x) + 2S_T(x)\eta, \tag{2.27}$$

where $S_T(x)$ is the side slope of the channel and $B(x)$ is its bottom width, see Figure 3. For the special case of a rectangular channel $S_T(x)$ equals zero. We can evaluate (2.20) and (2.21) to get

$$I_1(x,t) = D(x,t)^2 \left(\frac{B(x)}{2} + \frac{D(x,t)S_T(x)}{3} \right), \tag{2.28}$$

$$I_2(x,t) = D(x,t)^2 \left(\frac{1}{2}\frac{\partial B(x)}{\partial x} + \frac{D(x,t)}{3}\frac{\partial S_T(x)}{\partial x} \right). \tag{2.29}$$

Also, the wetted cross-sectional area $A(x,t)$ depends on the depth $D(x,t)$ of the water:

$$A(x,t) = D(x,t)B(x) + D(x,t)^2 S_T(x) \quad \text{and} \quad \frac{\partial A}{\partial D}(x,t) = B(x) + 2D(x,t)S_T(x). \tag{2.30}$$

Since $S_T(x) \geq 0$ and $B(x) > 0$ we can always determine the depth of the water $D(x,t)$ from $A(x,t)$:

$$D(x,t) = \begin{cases} \dfrac{A(x,t)}{B(x)}, & \text{if} \quad S_T(x) = 0 \\ -\dfrac{B(x)}{2S_T(x)} + \sqrt{\dfrac{B(x)^2}{4S_T(x)^2} + \dfrac{A(x,t)}{S_T(x)}}, & \text{if} \quad S_T(x) > 0. \end{cases} \tag{2.31}$$

Also, for a trapezoidal channel we have

$$T(x,D) = B(x) + 2D(x,t)S_T(x) \quad \text{and} \quad P(x,D) = B(x) + 2D(x,t)\sqrt{1 + S_T(x)^2}, \tag{2.32}$$

for the water surface width T and the wetted perimeter P. Therefore the system may be written in the conservative vector form

$$\mathbf{u}_t + \mathbf{f}_x = \mathbf{s}, \tag{2.33}$$

where

$$\begin{aligned} \mathbf{u} &= [A, Q]^T, & (2.34) \\ \mathbf{f} &= \left[Q, \frac{Q^2}{A} + g\left(\frac{D^2 B}{2} + \frac{D^3 S_T}{3}\right)\right]^T, & (2.35) \\ \mathbf{s} &= \left[0, gA(S_0 - S_f) + gD^2\left(\frac{1}{2}\frac{\partial B}{\partial x} + \frac{D}{3}\frac{\partial S_T}{\partial x}\right)\right]^T. & (2.36) \end{aligned}$$

The Jacobian of $\mathbf{f}$ is given by

$$\mathcal{A}(\mathbf{u}) = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} = \begin{bmatrix} 0 & 1 \\ c^2 - v^2 & 2v \end{bmatrix}, \tag{2.37}$$

where $v = \dfrac{Q}{A}$ is the average velocity and, using (2.30) and (2.32),

$$c^2 = g\frac{\partial I_1}{\partial A} = g\frac{(DB + D^2 S_T)\partial D}{\partial A} = g\frac{DB + D^2 S_T}{B + 2DS_T} = g\frac{A}{T}, \tag{2.38}$$

i.e. we get $c = \sqrt{\dfrac{gA}{T}}$ for the wave celerity. The eigenvalues of $\mathcal{A}(\mathbf{u})$ are given by

$$a_1 = v - c, \tag{2.39}$$

$$a_2 = v + c, \tag{2.40}$$

and the eigenvectors are

$$v_1 = \begin{bmatrix} 1 \\ v - c \end{bmatrix} \quad \text{and} \quad v_2 = \begin{bmatrix} 1 \\ v + c \end{bmatrix}. \tag{2.41}$$

Hence, $\mathcal{A}(\mathbf{u})$ has got 2 real eigenvalues and 2 linearly independent eigenvectors if $c \neq 0$. By definition the system is *hyperbolic* (see [13]), and also, since in the latter case the eigenvalues are distinct, *strictly hyperbolic*.

3 The Preissmann Box Scheme

The Preissmann Box Scheme is the standard method used by hydraulic engineers to describe one dimensional flow. It is a so-called cell-vertex scheme (Morton [21]) and, as a four-point finite-difference implicit scheme, unconditionally stable. However, the Box Scheme can only be used to model strictly subcritical or supercritical flow. We will show why it breaks down for transcritical flow.

3.1 Scalar Equation

We consider the scalar conservation law

$$u_t + f(u)_x = s(u, x), \tag{3.1}$$

and approximate u by a discrete set of values $u_j^n \approx u(x_j, t^n)$, which correspond to the function values on the rectangular grid for which the nodes are given by (x_j, t^n). We take the values of u and f at four corners of a computational box as it can be seen in Figure 4. Then we obtain the Box Scheme by replacing the partial derivatives in

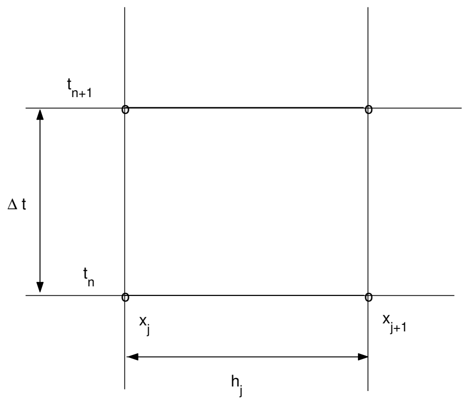

Figure 4: *The Preissmann Box Scheme stencil*

the equation (3.1) by its finite difference approximations,

$$\frac{\partial u}{\partial t} \approx \frac{u_{j+1}^{n+1} - u_{j+1}^n + u_j^{n+1} - u_j^n}{2\Delta t}, \tag{3.2}$$
$$\frac{\partial f}{\partial x} \approx \frac{\theta(f_{j+1}^{n+1} - f_j^{n+1}) + (1-\theta)(f_{j+1}^n - f_j^n)}{h_j}, \tag{3.3}$$

where $\theta \in [0,1]$. The grid points $x_0 < x_1 < \ldots < x_N$ are variably spaced with $h_j = x_{j+1} - x_j$ for all $j = 0, \ldots, N-1$. The spatial derivatives are centered in space and weighted in time according to the weighting factor θ. For $\theta = 0$ the scheme is fully explicit and for $\theta = 1$ it is fully implicit. The source term $s(u,x)$ is approximated by a weighted average of the four points in the stencil in Figure 4 with

$$s(u,x) \approx \frac{1}{2}\theta(s_{j+1}^{n+1} + s_j^{n+1}) + \frac{1}{2}(1-\theta)(s_{j+1}^n + s_j^n), \tag{3.4}$$

where $s_j^n = s(u_j^n, x_j)$. These approximations finally lead to the scheme

$$\begin{aligned} \frac{u_{j+1}^{n+1} - u_{j+1}^n}{2} + \frac{u_j^{n+1} - u_j^n}{2} + \frac{\Delta t}{h_j}\theta(f_{j+1}^{n+1} - f_j^{n+1}) + \frac{\Delta t}{h_j}(1-\theta)(f_{j+1}^n - f_j^n) \\ - \Delta t \left(\frac{1}{2}\theta(s_{j+1}^{n+1} + s_j^{n+1}) + \frac{1}{2}(1-\theta)(s_{j+1}^n + s_j^n) \right) = 0. \end{aligned} \tag{3.5}$$

for $j = 0, \ldots, N-1$.

3.2 Systems of Equations

We consider the differential form of the St Venant equations

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{s}(\mathbf{u}, x), \tag{3.6}$$

where $\mathbf{u} = [A, Q]^T$, and apply the Box Scheme (3.5) in vector form, which gives

$$\begin{aligned} \frac{\mathbf{u}_{j+1}^{n+1} - \mathbf{u}_{j+1}^n}{2} + \frac{\mathbf{u}_j^{n+1} - \mathbf{u}_j^n}{2} + \lambda_j\theta(\mathbf{f}_{j+1}^{n+1} - \mathbf{f}_j^{n+1}) + \lambda_j(1-\theta)(\mathbf{f}_{j+1}^n - \mathbf{f}_j^n) \\ - \Delta t \left(\frac{1}{2}\theta(\mathbf{s}_{j+1}^{n+1} + \mathbf{s}_j^{n+1}) + \frac{1}{2}(1-\theta)(\mathbf{s}_{j+1}^n + \mathbf{s}_j^n) \right) = \mathbf{0}, \end{aligned} \tag{3.7}$$

for all $j = 0, \ldots, N-1$, where $\lambda_j = \dfrac{\Delta t}{h_j}$ and $h_j = x_{j+1} - x_j$. The vectors $\mathbf{u}, \mathbf{f}$ and s are taken from (2.34)-(2.36). Therefore we may implement the Box Scheme for a general trapezoidal channel including friction forces and variable width.

3.3 Implementation

In order to implement the Box Scheme we consider a finite domain $x \in [0, L]$ and use $N+1$ grid points $[x_0, \dots, x_N]$ which are variably spaced with distance h_j. This variable spacing allows us to respond to changes in channel width and bed slope and helps to approximate the depth of the water more exactly. It might also be useful for shock recovery. The implementation procedure is described in [9]. Since there are two unknowns A_j and Q_j at each node x_j, we have a total number of $2N+2$ unknowns. By defining the left-hand side of (3.7) to be the cell residual $\mathbf{R}_{j+\frac{1}{2}}$ of each of the N cells $[x_j, x_{j+1}],\ j = 0, \dots, N-1$, we get $2N$ equations. Thus, we need two more equations, which are supplied by the boundary conditions. For subcritical flow the discharge Q is usually prescribed upstream and the height h downstream. Both Q and h are prescribed at the upstream boundary for supercritical flow.

Since the St Venant equations are nonlinear we need a nonlinear iteration technique, to solve

$$\mathbf{R}_{j+\frac{1}{2}} = 0, \quad \forall j = 0, \dots, N-1. \tag{3.8}$$

We also need an efficient method, since *within* each time step we have to iterate successively. We use Newton's Method for systems, which is described in [2], in order to solve (3.8). Suppose the solution is known up to time level t^n and we want to find the solution at time level t^{n+1}. Let $\mathbf{u}^{(k)}$ be the estimate of the solution $\mathbf{u}^{n+1}$ after k iterations. The residual for this estimate is

$$\begin{aligned}\mathbf{R}^{(k)}_{j+\frac{1}{2}} = \frac{\mathbf{u}^{(k)}_{j+1} - \mathbf{u}^n_{j+1}}{2} + \frac{\mathbf{u}^{(k)}_{j} - \mathbf{u}^n_{j}}{2} + \lambda_j\theta(\mathbf{f}^{(k)}_{j+1} - \mathbf{f}^{(k)}_{j}) + \lambda_j(1-\theta)(\mathbf{f}^n_{j+1} - \mathbf{f}^n_j) \\ - \Delta t\left(\frac{1}{2}\theta(\mathbf{s}^{(k)}_{j+1} + \mathbf{s}^{(k)}_{j}) + \frac{1}{2}(1-\theta)(\mathbf{s}^n_{j+1} + \mathbf{s}^n_j)\right).\end{aligned} \tag{3.9}$$

The next iterate $\mathbf{u}^{(k+1)}$ is obtained by solving the Newton system

$$\begin{aligned}\frac{1}{2}(I + 2\lambda_j\theta\mathcal{A}^{(k)}_{j+1})\Delta\mathbf{u}^{(k)}_{j+1} + \frac{1}{2}(I - 2\lambda_j\theta\mathcal{A}^{(k)}_{j})\Delta\mathbf{u}^{(k)}_{j} \\ - \frac{1}{2}\theta\Delta t\mathcal{S}^{(k)}_{j+1}\Delta\mathbf{u}^{(k)}_{j+1} - \frac{1}{2}\theta\Delta t\mathcal{S}^{(k)}_{j}\Delta\mathbf{u}^{(k)}_{j} = -\mathbf{R}^{(k)}_{j+\frac{1}{2}},\end{aligned} \tag{3.10}$$

where $\mathcal{A} = \dfrac{\partial \mathbf{f}}{\partial \mathbf{u}}$, $\mathcal{S} = \dfrac{\partial \mathbf{s}}{\partial \mathbf{u}}$ and $j = 0, \ldots, N-1$. For the Jacobian we assume that

$$\frac{\partial B}{\partial x} = 0 \quad \text{and} \quad \frac{\partial S_T}{\partial x} = 0, \tag{3.11}$$

and therefore

$$\mathcal{A} = \begin{bmatrix} 0 & 1 \\ c^2 - v^2 & 2v \end{bmatrix} \quad \text{and} \quad \mathcal{S} = \begin{bmatrix} 0 & 0 \\ g\left(S_0 + S_f\left(\frac{7}{3} - \frac{8A}{3TP}\sqrt{1+S_T^2}\right)\right) & -\frac{2gA}{Q}S_f \end{bmatrix}, \tag{3.12}$$

where the wave celerity c and the average velocity v are given by

$$c = \sqrt{\frac{gA}{T}} \quad \text{and} \quad v = \frac{Q}{A}. \tag{3.13}$$

The system, we have to solve for each state $\mathbf{u}^{(k)}$, may then be written in block-tridiagonal form. If we multiply by 2 and, just for the purpose of a simpler notation, omit $\mathcal{S}$, the system may be written in the form

$$M\mathbf{W} = \mathbf{Z}, \tag{3.14}$$

where

$$M = \begin{bmatrix} \alpha_0 \quad \beta_0 & \mathbf{0} & \cdots & \cdots & \cdots \\ I - 2\theta\lambda_0\mathcal{A}_0 & I + 2\theta\lambda_0\mathcal{A}_1 & \mathbf{0} & \cdots & \cdots \\ \mathbf{0} & I - 2\theta\lambda_1\mathcal{A}_1 & I + 2\theta\lambda_1\mathcal{A}_2 & \mathbf{0} & \cdots \\ \cdots & \cdots & \ddots & \ddots & \cdots \\ \cdots & \mathbf{0} & I - 2\theta\lambda_{N-2}\mathcal{A}_{N-2} & I + 2\theta\lambda_{N-2}\mathcal{A}_{N-1} & \mathbf{0} \\ \cdots & \cdots & \mathbf{0} & I - 2\theta\lambda_{N-1}\mathcal{A}_{N-1} & I + 2\theta\lambda_{N-1}\mathcal{A}_N \\ \cdots & \cdots & \cdots & \mathbf{0} & \alpha_N \quad \beta_N \end{bmatrix} \tag{3.15}$$

with the Jacobian $\mathcal{A}_j$ evaluated at x_j, $j = 0, \ldots, N$, and

$$\mathbf{W} = \begin{bmatrix} \Delta\mathbf{u}_0^{(k)} & \Delta\mathbf{u}_1^{(k)} & \cdots & \Delta\mathbf{u}_{N-1}^{(k)} & \Delta\mathbf{u}_N^{(k)} \end{bmatrix}^T, \tag{3.16}$$

$$\mathbf{Z} = \begin{bmatrix} \gamma_0 & -2\mathbf{R}_{\frac{1}{2}} & -2\mathbf{R}_{\frac{3}{2}} & \cdots & -2\mathbf{R}_{N-\frac{3}{2}} & -2\mathbf{R}_{N-\frac{1}{2}} & \gamma_N \end{bmatrix}^T. \tag{3.17}$$

The first and the last equation

$$\begin{aligned} \alpha_0 \Delta A_0 + \beta_0 \Delta Q_0 &= \gamma_0, & (3.18) \\ \alpha_N \Delta A_N + \beta_N \Delta Q_N &= \gamma_N, & (3.19) \end{aligned}$$

represent the boundary conditions at inflow and outflow. The number of boundary conditions at inflow and outflow is determined by the kind of flow present.

Once the new iterate $\mathbf{u}^{(k+1)}$ is calculated using the Newton update

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \Delta \mathbf{u}^{(k)}, \tag{3.20}$$

the whole process is repeated, until the stopping criterion,

$$\frac{\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_1}{\|\mathbf{u}^{(k)}\|_1} < \text{tol}, \tag{3.21}$$

is satisfied, where the tolerance tol is a small positive number or until a fixed number of iterations has been carried out.

Note, that if (3.11) is not satisfied or by omitting $\mathcal{S}$ in the Jacobian, we only get a Quasi-Newton method.

3.4 Solution Procedure

We consider available methods in order to solve the system of linearised equations. Standard matrix inversion techniques usually need a number of operations which is proportional to N^3 ,or, at best N^2 (see, for example [5]), where N is the number of grid points. In order to solve the block-tridiagonal Newton system (3.14) with 2×2 blocks, we may use the Thomas algorithm, a non-pivoting Gaussian elimination for tridiagonal matrices which can be extended to block-tridiagonal matrices. The number of operations is proportional to the number of grid points N ([1]). This method is identical to successively applying the double sweep algorithm and described for an easier system in [20] and in [28].

The tridiagonal system of linear equations may be written in the form

$$A_i\mathbf{W}_{i-1} + D_i\mathbf{W}_i + C_i\mathbf{W}_{i+1} = \mathbf{Z}_i, \quad i = 0, \ldots, N, \tag{3.22}$$

where A_i, D_i and C_i are 2x2 matrices and $\mathbf{W}_i = [\Delta A_i, \Delta Q_i]^T$. Using the boundary conditions at inflow and outflow we get

$$A_0 = 0, \quad \text{and} \quad A_j = \begin{bmatrix} -2\theta\lambda_{j-1}(c_{j-1}^2 - v_{j-1}^2) & 1 - 4\theta\lambda_{j-1}v_{j-1} \\ 0 & 0 \end{bmatrix}, \quad j = 1, \ldots N, \tag{3.23}$$

for the subdiagonal elements,

$$C_N = 0, \quad \text{and} \quad C_j = \begin{bmatrix} 0 & 0 \\ 1 & 2\theta\lambda_j \end{bmatrix}, \quad j = 0, \ldots N-1, \tag{3.24}$$

for the superdiagonal elements and

$$D_0 = \begin{bmatrix} \alpha_0 & \beta_0 \\ 1 & -2\theta\lambda_0 \end{bmatrix}, \quad D_N = \begin{bmatrix} 2\theta\lambda_N(c_N^2 - v_N^2) & 1 + 4\theta\lambda_N v_N \\ \alpha_N & \beta_N \end{bmatrix} \tag{3.25}$$

and

$$D_j = \begin{bmatrix} 2\theta\lambda_j(c_j^2 - v_j^2) & 1 + 4\theta\lambda_j v_j \\ 1 & -2\theta\lambda_j \end{bmatrix}, \quad j = 1, \ldots N-1, \tag{3.26}$$

for the diagonal block entries. Direct solution gives the backwards recursion

$$\mathbf{W}_N = \mathbf{F}_N, \tag{3.27}$$

$$\mathbf{W}_j = E_j\mathbf{W}_{j+1} + \mathbf{F}_j, \quad j = N-1, \ldots, 0, \tag{3.28}$$

where E_j and $\mathbf{F}_j$ are given by the forward recursion

$$E_j = -(D_j + A_jE_{j-1})^{-1}C_j, \quad j = 0, \ldots, N-1, \tag{3.29}$$

$$\mathbf{F}_j = (D_j + A_jE_{j-1})^{-1}(\mathbf{Z}_j - A_j\mathbf{F}_{j-1}), \quad j = 0, \ldots, N, \tag{3.30}$$

and $\mathbf{F}_{-1} = 0$, $E_{-1} = 0$. This technique is has a very similar structure to the simple Marching Scheme, described in [25] and is a stable procedure if the inverses exist and if

$$\|E_j\| \leq 1, \quad \forall j. \tag{3.31}$$

Diagonal dominance is a sufficient condition to guarantee (3.31) as stated for the scalar case in Morton and Mayers [24]. The generalisation for the case of block-tridiagonal systems is

$$(\|A_j\| + \|C_j\|)\|D_j^{-1}\| \leq 1 \quad \forall j. \tag{3.32}$$

In section 6.4 we will show with a similar analysis as it was done in Morton [20], that our Thomas Algorithm is well-conditioned. The described algorithm is easier and faster than setting up and solving the whole system, since less storage is required and only 2x2 matrices have to be inverted.

3.5 Overall Algorithm Description

In order to implement the above algorithm, we need to set up the boundary conditions as well as the initial conditions at the nodes. For the cross-sectional area A_j and and the discharge Q_j we take constant initial conditions, subject to the boundary conditions. Then, for each time step, we perform the following algorithm, until the steady state is reached:

1. set up A_j, D_j, C_j and $\mathbf{Z}_j$ for all $j = 0, \ldots, N$ using (3.23)-(3.26);
2. calculate E_j, $j = 0, \ldots, N-1$, and $\mathbf{F}_j$, $j = 0, \ldots, N$ by the forward recursions (3.29) and (3.30);
3. solve the system by the Thomas Algorithm (3.27) and (3.28);
4. calculate the new iterate using (3.20);
5. check, if stopping criterion (3.21) is satisfied. If so, then move on to the next time level, otherwise start with 1. again.

Hence, for each time step, we have to do a Newton iteration.

3.6 Transcritical Flow

The term transcritical flow denotes the existence of both subcritical and supercritical flow regions in an open-channel system. It is similar to transonic flow in gas dynamics and occurs in irrigation canals especially on steep slopes or rapidly changing channel width.

In order to characterise the flow, the eigenvalues of the Jacobian (2.37),

$$a_1 = v - c, \tag{3.33}$$

$$a_2 = v + c, \tag{3.34}$$

are important. As stated in [9] the eigenvalues can be thought of as the velocities at which disturbances propagate. When the eigenvalues have opposite signs, disturbances travel both upstream and downstream. When the eigenvalues have the same sign, no information is carried upstream and disturbances can only travel downstream. In the first case, the flow is said to be subcritical, in the latter case it is supercritical. By calculating the *Froude number* (see [7]),

$$F = \frac{|v|}{c}, \quad \text{with} \quad c = \sqrt{\frac{gA}{T}} \tag{3.35}$$

similar to the Mach number in gas flow, the flow may be determined to be subcritical or supercritical as F is less or greater than unity. Some authors refer tranquil and rapid flow instead [3]. From (3.35) we clearly see, that for a slowly flowing river we always have subcritical flow. We also find that supercritical flow occurs, if the velocity of the stream is increasing. This happens if the bed slope steepens, or, if water flows through a narrow region.

A typical situation, that arises in a river is the following: The flow is subcritical at inflow and at outflow, i.e. the bed slope is very gentle. If the velocity of the stream increases, we will have a region of supercritical flow. Typically, flow changes smoothly to supercritical flow at the so called *sonic point* and returns to subcritical flow via a *shock* or *hydraulic jump*.

As we will explain more detailed in section 7.1, setting the cell residuals to zero won't work in this subcritical - supercritical - subcritical case. It turns out, that there is too little information at the sonic point, and too much information at the shock.

The problem is therefore locally ill-posed and cannot be solved.

Hence, flows in parts of the river with steep bed slope or rapidly changing channel width cannot be resolved. We will see some examples for this limitation in the following 2 sections.

4 Model Problems

We need to create test problems with a known analytical solution, in order to check our algorithms. In this section we describe how to construct such test problems and we introduce some problems to which we will later apply our algorithms.

4.1 Steady Open Channel Test Problems

In [15] and [16] a method is described, that allows the construction of test problems, where in each case the exact solution for the steady St Venant equations is known. Under steady state conditions it is assumed that the depth $D = D(x)$ and the discharge $Q = Q(x)$. Furthermore we use a trapezoidal channel with $I_2 = 0$, i.e. the side slope S_T and the channel width B are constant. Then equations (2.22) and (2.23) reduce to

$$\begin{aligned} \frac{\partial Q}{\partial x} &= 0, && (4.1) \\ \frac{\partial}{\partial x}\left(\frac{\beta Q^2}{A} + g\left(\frac{D^2 B}{2} + \frac{D^3 S_T}{3}\right)\right) &= gA(S_0 - S_f). && (4.2) \end{aligned}$$

From the first equation, Q is clearly constant along the length of the channel. Differentiating the momentum term in the second equation yields

$$\left(1 - \frac{Q^2 T}{gA^3}\right)\frac{dD}{dx} = S_0 - S_f. \tag{4.3}$$

Hence, the bed slope function $S_0(x)$ can be deduced from (4.3). The usual approach is now an inverse one. We choose a smooth depth function $D = D(x)$ and use (4.3) in order to calculate the bed slope S_0. The analytic solution to this steady state problem is then given by $D = D(x)$. Constructing test problems with a non-smooth analytic solution is similar, but we have to take care about the hydraulic jump. The hypothetical flow must be physically allowable. A procedure for achieving this is described in [16].

We are going to study the *unsteady case*, but still, the steady solution is important. On the one hand, if constant boundary conditions are imposed and we let $t \to \infty$, then the flow converges to a steady state, for which the analytical solution is known. On the other hand steady flows are often required as initial data for unsteady simulations. We therefore state some steady test problems.

4.2 Subcritical Problem with Small Froude Number (P1)

The first problem, that we consider is a trapezoidal channel of length $1\,km$, width $B = 10\,m$ and side slope $S_T = 1$. We impose a discharge of $Q = 20\,m^3/s$ at inflow and a depth of $D_{\text{out}} = 1.112299\,m$. The flow is subcritical at inflow and outflow. As Manning roughness coefficient we choose $n = 0.02$. Using (4.3) the slope of the channel is then given by

$$S_0(x) = \left(1 - \frac{Q^2 T}{g A^3}\right) D'(x) + \frac{Q^2 n^2 P^{\frac{4}{3}}}{A^{\frac{10}{3}}}, \tag{4.4}$$

where P, T and A are chosen according to (2.32) and (2.30). The depth $D(x)$ of the water for the steady problem is given by

$$D(x) = \left(\frac{4}{g}\right)^{\frac{1}{3}} \left(\frac{3}{2} - \frac{2}{5}\exp\left(-144\left(\frac{x}{1000} - \frac{1}{2}\right)^2\right)\right). \tag{4.5}$$

The solution for the steady state is shown in Figures 5 and 6.

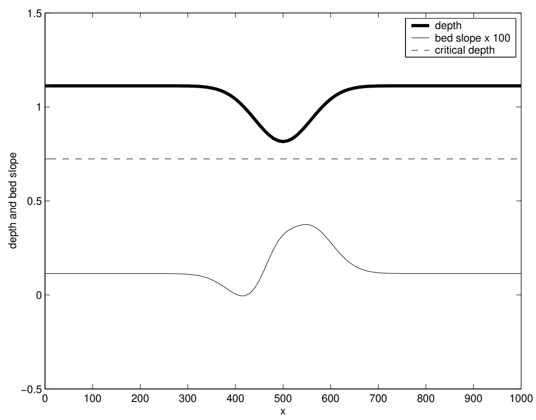

Figure 5: *Depth and bed slope* $(P1)$

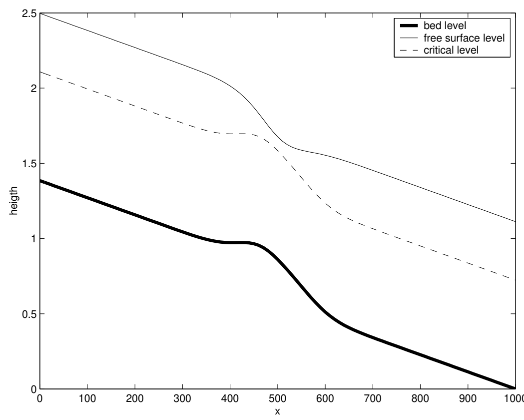

Figure 6: *Bed level and surface level* $(P1)$

4.3 Near-critical Problem (P2)

We choose the same channel geometry as in the previous problem (P1), but with depth $D_{\text{out}} = 0.748323\,m$ at outflow. As Manning roughness coefficient we choose $n = 0.03$. The depth of the water for the steady problem is given by

$$D(x) = \left(\frac{4}{g}\right)^{\frac{1}{3}} \left(1 + \frac{1}{2}\exp\left(-16\left(\frac{x}{1000} - \frac{1}{2}\right)^2\right)\right), \tag{4.6}$$

and the slope is again calculated by (4.4). The solution for the steady state is shown in Figures 7 and 8. We include this problem, because it is close to being transcritical

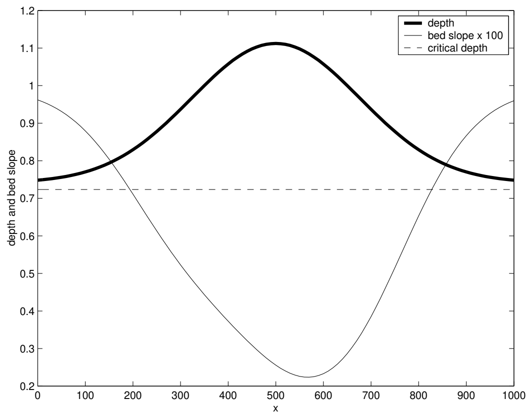

Figure 7: *Depth and bed slope* $(P2)$

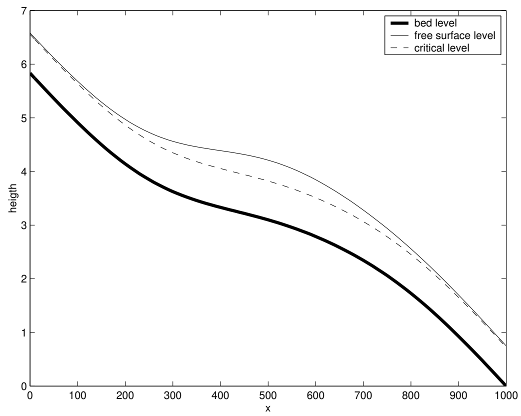

Figure 8: *Bed level and surface level* $(P2)$

at both ends of the channel.

4.4 Transcritical Problem Involving a Hydraulic Jump (P3)

As a test problem for transcritical flow we take an example from [16]. Again, we choose a trapezoidal channel of length $1\,km$, width $B = 10\,m$ and side slope $S_T = 1$. The inlet discharge is $Q = 20\,m^3/s$. Now, flow is subcritical at inflow and at outflow with depth $D_{\text{out}} = 1.349963\,m$, but, due to the steepening bed slope, it changes smoothly to supercritical flow and returns to subcritical flow via a hydraulic jump. As Manning roughness coefficient we choose $n = 0.02$. Then the bed slope of the channel is given by (4.4), where the depth of the water is

$$D(x) = \begin{cases} 0.723449\left(1-\tanh\left(\frac{x}{1000}-\frac{3}{10}\right)\right), & 0 \le x \le 300, \\ 0.723449\left(1-\frac{1}{6}\tanh\left(6\left(\frac{x}{1000}-\frac{3}{10}\right)\right)\right), & 300 < x \le 600, \\ \frac{3}{4}+\sum_{k=1}^{3} a_k \exp\left(-20k\left(\frac{x}{1000}-\frac{3}{5}\right)\right)+\frac{3}{5}\exp\left(\frac{x}{1000}-1\right), & 600 < x \le 1000, \end{cases} \tag{4.7}$$

with $a_1 = -0.111051$, $a_2 = 0.026876$ and $a_3 = -0.217567$. The analytic solution for the steady state is shown in Figures 9 and 10.

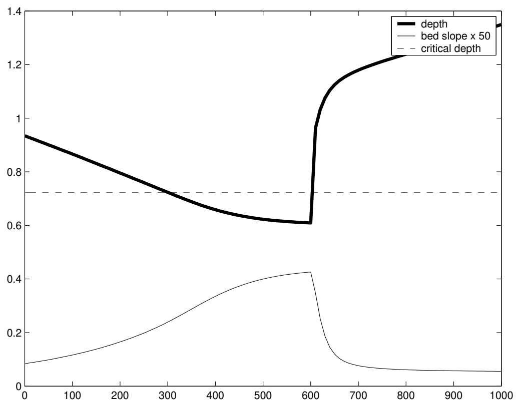

Figure 9: *Depth and bed slope* $(P3)$

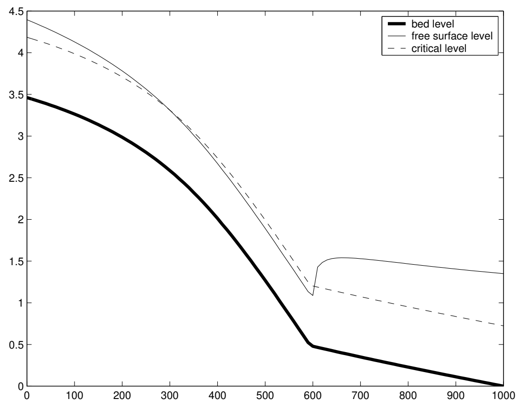

Figure 10: *Bed level and surface level* $(P3)$

4.5 Transcritical Problem with Changing Channel Width (P4)

This test problem is not constructed using the method described in section 4.1. Instead, it is taken from Skeels' thesis [29, page 144]. We take a rectangular channel of length $10\,km$, and smoothly changing width from $B = 10\,m$ to $B = 5\,m$ and back to $B = 10\,m$ again. The width is obtained via a cubic polynomial,

$$B(x) = \frac{2}{25}\left(\frac{x}{1000}\right)^3 - \frac{3}{5}\left(\frac{x}{1000}\right)^2 + 10, \quad x = 0, \ldots, 5000, \tag{4.8}$$

mirrored in the line $x = 5000$. The upstream boundary condition is constant $Q = 20\,m^3/s$ and at the downstream boundary we impose the depth $D_{\text{out}} = 1.3\,m$. The bed slope is constant $S_0(x) = 0.002$ except between 4500 and 5500 metres, where the bed slope is doubled (P4a). As Manning roughness coefficient we choose $n = 0.03$. The channel bed and the channel width are illustrated in Figures 11 and 12.

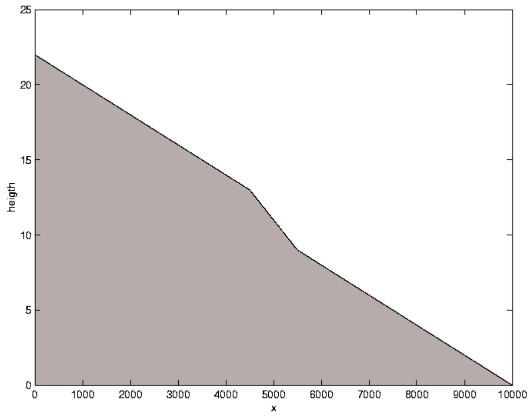

Figure 11: *Channel bed* $(P4)$

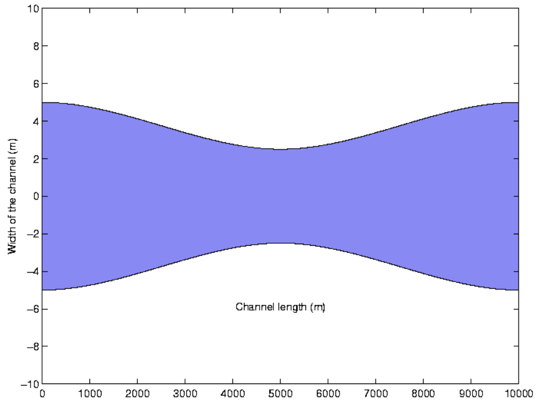

Figure 12: *Channel width* $(P4)$

We can modify problem (P4a), which is subcritical, in order to investigate different situations. By increasing the bed slope or by decreasing the channel width, the average velocity v and therefore the Froude Number (3.35) increases. Hence, the problem becomes transcritical, i.e. subcritical with an interior supercritical region. This variation leads to problem (P4b): The bed slope is constant $S_0(x) = 0.002$ except between 4500 and 5500 metres, where the bed slope is $S_0(x) = 0.012$, i.e. six times steeper. Further modification leads to problem (P4c): The bed slope is constant $S_0(x) = 0.002$ but between 4500 and 5500 metres it is $S_0(x) = 0.04$, i.e. twenty times steeper.

5 Steady State Solutions to Model Problems

We solve our unsteady test problems with the algorithm described above, i.e. by driving the cell residuals to zero using Newton's Method. We monitor the maximum Froude Number for each time state t^n,

$$F = \max_{j=0,\ldots,N} \frac{|v_j|}{c_j}, \tag{5.1}$$

and also the maximum CFL number, which is

$$\nu_{max} = \max\{\nu_j^n : j = 0, \ldots, N;\ n = 0, \ldots, n_T\}, \tag{5.2}$$

where t^{n_T} is the final time, i.e. the time when the steady state is reached. The CFL number ν_j^n of the system at (x_j, t^n) is defined by

$$\nu_j^n = \lambda_j \rho(\mathcal{A}(\mathbf{u}_j^n)), \tag{5.3}$$

where $\rho(.)$ denotes the spectral radius. By keeping the boundary conditions constant, flow is allowed to approach the steady state. We will see that the Box Scheme in its original implementation does not solve transcritical problems.

5.1 Subcritical Problem with Small Froude Number (P1)

For this problem we use Newton's Method, i.e. the full Jacobian. We use constants for discharge Q and depth D, which match the boundary conditions, as initial values. With $\theta = \frac{2}{3}$, $\Delta x = 10$, i.e. $N = 100$, $\Delta t = 10$ and a tolerance of 10^{-10}, the resulting depth for the steady state (after $t = 900$) may be seen in Figure 13. Initially the maximum Froude Number is 0.5138, the maximum Froude Number for the steady state is 0.8308. Approximations to the Froude Number for the steady state may be seen in Figure 14. The maximum CFL number in this case is $\nu_{max} = 5.1238$. Convergence at each time step takes about $3-5$ iterations. We can increase the time step and therefore the maximum CFL number, but we need more iterations for convergence at each time step. If we omit the derivative of the source term $\mathcal{S}$ in the Jacobian, we have to reduce the time step to about $\Delta t = 10$, and convergence takes about $7 - 10$

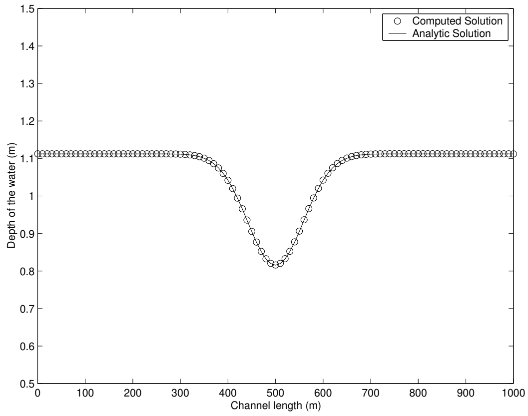

Figure 13: *Subcritical problem with small Froude Number, steady state ($P1$)*

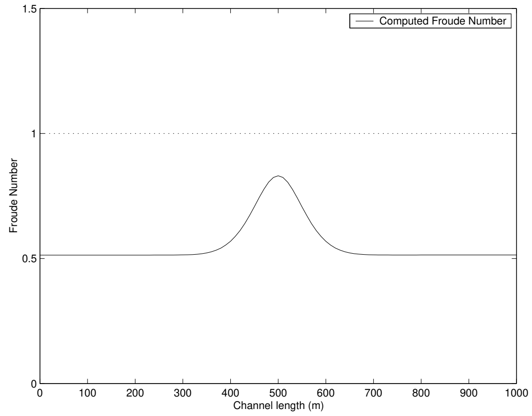

Figure 14: *Approximations for Froude Number for subcritical problem ($P1$)*

iterations. If Δt is chosen too large, we get no convergence after the first time step. This behaviour, due to Newton's Method, will be explained in a later section.

We want to use this problem to check the order of convergence for the Box Scheme. On that score we set $\theta = \frac{1}{2}$ and assume that the error $e(\Delta x)$ satisfies $e(\Delta x) = C\Delta x^{\beta}$ for some constant C. Then we can calculate β using

$$\beta(\Delta x) = \frac{\ln \frac{e(2\Delta x)}{e(\Delta x)}}{\ln 2}. \tag{5.4}$$

Several computations have been carried out and the error $e(\Delta x) = \|D_{\text{anal}} - D_{\text{num}}\|_{\infty}$ has been calculated for problem (P1). Table 1 suggests $\mathcal{O}(\Delta x^2)$ convergence for the

Δx	N	$e(\Delta x) = \|D_{\text{ANAL}} - D_{\text{NUM}}\|_{\infty}$	$\beta(\Delta x)$
40	25	0.00868	-
20	50	0.002997	1.53
10	100	$9.42988 \cdot 10^{-4}$	1.68

Table 1: *Convergence rate*

Box Scheme. In section 6 we will see that there is very little dissipation for $\theta = \frac{1}{2}$ and oscillations occur, which warp the 2nd order convergence of the Box Scheme. In order to check the robustness of the method, we can also try $\theta = 1$, which should almost omit time-stepping. Using $N = 100$ and $\Delta t = 900$, we get indeed a very adequate solution for the steady state, after just one time-step and 8 iterations.

5.2 Near-critical Problem (P2)

Again, we use the full Jacobian and $\theta = \frac{2}{3}$, $\Delta x = 10$, $\Delta t = 10$ and a tolerance of 10^{-10}. The initial values are constants, matching the boundary conditions. Small period $2\Delta x$ oscillations occur at the boundary after the first time step. The steady state solution can be seen in Figure 16. Initially the maximum Froude Number is

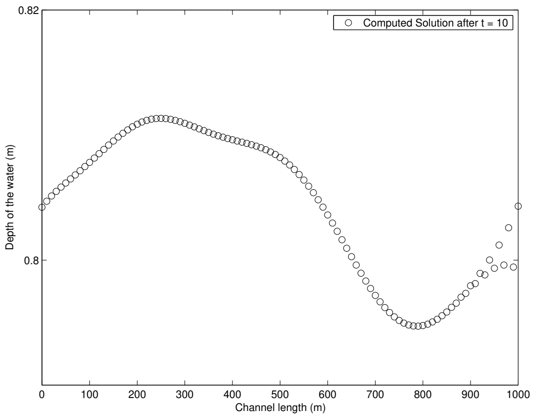

Figure 15: Near-critical problem after one time-step, $t = 10$ ($P2$)

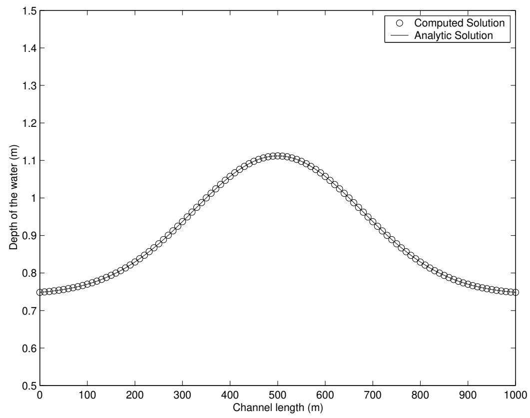

Figure 16: Near-critical problem at steady state ($P2$)

0.9492, the maximum Froude Number for the steady state is 0.9492, too. The maximum CFL number in this case is $\nu_{max} = 5.1065$. Convergence takes about $4 - 5$ iterations for each time step. After the first time step, the oscillations on the right hand boundary are largest (see Figure 15), due to the Froude number being close to unity. The oscillations die out during the iterations and cannot be seen in the steady state solution. Again, the same result is obtained by omitting the derivative of the source term $\mathcal{S}$ in the Jacobian, with $\Delta t = 10$, $15 - 29$ iterations for convergence for each time step. If Δt is chosen too large, we get no convergence after the first time step.

5.3 Transcritical Problem Involving a Hydraulic Jump (P3)

Using $\theta = \frac{2}{3}$, $\Delta x = 10$, $\Delta t = 1$, constant initial values and a tolerance of 10^{-10} again, convergence takes about 4 iterations for each time step. The maximum Froude Number for the initial state is 0.3749 and it increases during the iterations. As soon as the Froude Number approaches unity, the number of iterations for each time step increases and period $2\Delta x$ oscillations occur in the supercritical region, especially near the shock. In Figure 17 we can see the result for maximum Froude Number 1.0833 and in Figure 18 the result for maximum Froude Number 1.1735. Shortly after that time, the scheme breaks down, and we get a completely wrong solution. The steady state solution cannot even be approached, because it is transcritical. If Δt is

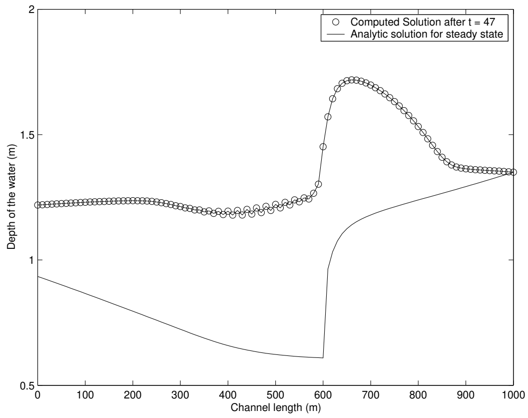

Figure 17: *Transcritical problem at time $t = 47$ ($P3$)*

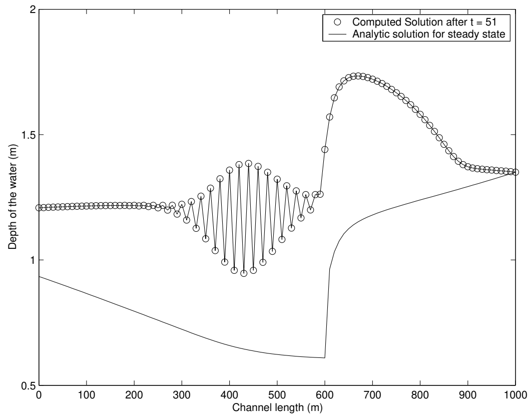

Figure 18: *Transcritical problem at time $t = 51$ ($P3$)*

chosen too large, we get no convergence after the first time step. The restriction on the time step is stronger than in the previous cases.

5.4 Transcritical Problem with Changing Channel Width (P4)

For the problem with changing channel width we just take a Quasi-Newton method. We calculate the Jacobian by assuming that $\frac{\partial B}{\partial x}$ can be neglected. Since $\max\left(\frac{\partial B}{\partial x}\right) = 0.0015$ this is justifiable. We use $\theta = \frac{2}{3}$, $\Delta x = 100$, i.e. $N = 100$ cells and $\Delta t = 100$, a tolerance of 10^{-8} and constant initial conditions matching the boundary values. After $t = 8000$ the steady state is approached. The result we get for the depth function can be seen in Figure 19. Initially the maximum Froude Number is 0.8616, the maximum Froude Number for the steady state is 0.5157. In Figure 20 we can see the approximation of the Froude Number for the steady state, which clearly indicates the subcritical case. The maximum CFL number is $\nu_{max} = 6.6481$ and convergence at each time step takes about $3 - 6$ Newton iterations. We can increase the time step and therefore the maximum CFL number, but we need slightly more iterations for convergence. If we omit the source term $\mathcal{S}$ in the Jacobian entirely, we still get convergence, but only if we reduce the time-step, for example to $\Delta t = 10$, and convergence takes much longer ($7 - 16$ iterations).

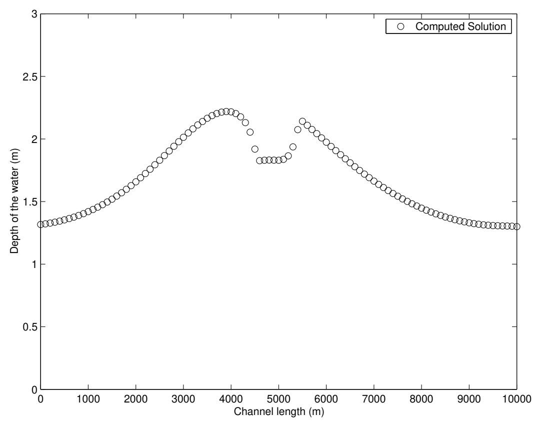

Figure 19: *Subcritical problem with changing channel width at steady state* ($P4a$)

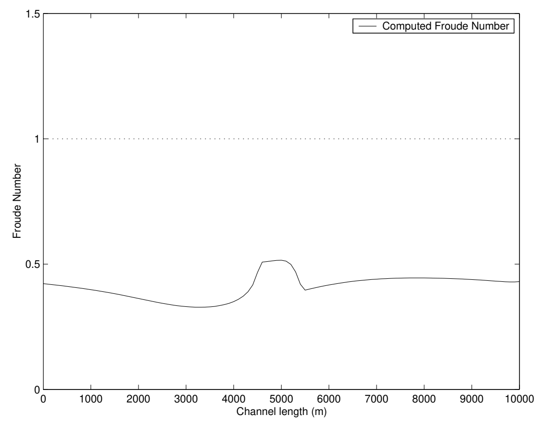

Figure 20: *Approximations to Froude Number for subcritical problem at steady state* ($P4a$)

In Figure 21, we see the result for the same problem but with a six times steeper bed slope between $x = 4500$ and $x = 5500$. The problem becomes critical at the steady state. Figure 22 shows Froude Number approximations for the steady state. From that plot we can see that the problem is near-critical with a Froude Number close to unity at the steady state. Just two nodes are in the supercritical region and the problem can still be solved by our unmodified Box Scheme implementation (see [29]). The maximum CFL number is $\nu_{max} = 7.0872$. Oscillations occur in the critical region, especially near the shock.

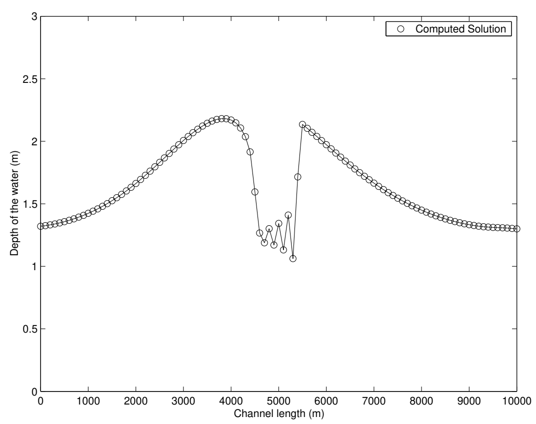

Figure 21: *Near-critical problem with changing channel width, steady state* ($P4b$)

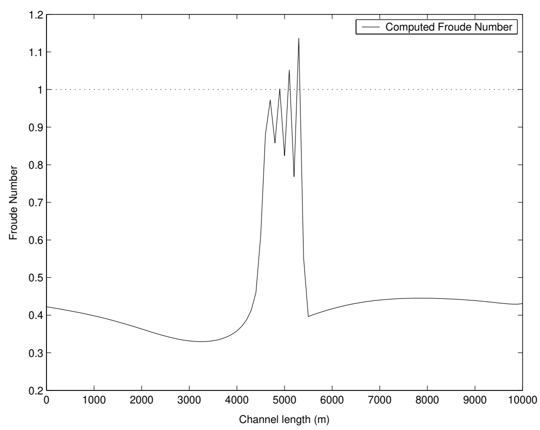

Figure 22: *Approximations to Froude Number for near-critical problem, steady state* ($P4b$)

For the same problem with a twenty times steeper bed slope (P4c) between $x = 4500$

and $x = 5500$ the flow becomes supercritical in that particular region. For $t = 0$ the problem is wholly subcritical, but after running the algorithm for a certain time $t > 0$, the unsteady solution becomes supercritical between $x = 4500$ and $x = 5500$ and and our unmodified Box Scheme implementation breaks down.

6 Accuracy, Stability and time-step Constraint

In this section we investigate the accuracy and stability of the Box Scheme applied to systems of equations. By this means we will also find some explanation, why the Box Scheme cannot be applied to transcritical flow.

6.1 Accuracy of the Box Scheme

We use Taylor's series expansion (see [6]) about $\left(x_{j+\frac{1}{2}}, t^{n+\frac{1}{2}}\right)$ in order to investigate the accuracy of the Box Scheme (3.5) for a regular sized mesh. By using the homogeneous equation $u_t + f_x = 0$ only, the truncation error (see [22]) of the Box Scheme may be written as

$$T_j^n = u_{txx}\frac{\Delta x^2}{8} + u_{ttt}\frac{\Delta t^2}{24} + f_{ttx}\frac{\Delta t^2}{8} + f_{xxx}\frac{\Delta x^2}{24} - f_{xt}\frac{\Delta t}{2} + \theta(f_{xt}\Delta t + \mathcal{O}(\Delta t^3, \Delta x^3)), \quad (6.1)$$

where the derivatives are evaluated at $\left(x_{j+\frac{1}{2}}, t^{n+\frac{1}{2}}\right)$. Therefore the Box Scheme is first order accurate for general values of θ. If we choose $\theta = \frac{1}{2}$, then the truncation error becomes

$$T_j^n = u_{txx}\frac{\Delta x^2}{12} - u_{ttt}\frac{\Delta t^2}{12} = \mathcal{O}(\Delta x^2, \Delta t^2), \quad (6.2)$$

and the Box Scheme is second order accurate in space and time for this special value of θ. In practice $\theta = \frac{1}{2} + \mathcal{O}(\Delta t)$ is commonly used in order to get second order accuracy.

6.2 Stability of the Box Scheme via Fourier Analysis

Consider the stability of the Box Scheme applied to the linearised form of the homogeneous St Venant equations (3.6),

$$\mathbf{u}_t + \mathcal{A}\mathbf{u}_x = 0, \quad (6.3)$$

where $\mathcal{A}$ is constant and given by (2.37). This system may be decoupled by introducing characteristic variables $\mathbf{w} = [w^1, w^2]^T$ (see [13]), with $\mathbf{w} = V^{-1}\mathbf{u}$, where V is the matrix of eigenvectors (2.41) of $\mathcal{A}$. Then, the system (6.3) can be written as two linear scalar equations,

$$w_t^1 + (v-c)w_x^1 = 0, \tag{6.4}$$

$$w_t^2 + (v+c)w_x^2 = 0. \tag{6.5}$$

Now, in this particular linear constant coefficient case, we can apply Fourier analysis (see Morton and Mayers [24]) individually to each of those two equations. For convenience, we write them in a general form,

$$w_t + aw_x = 0, \tag{6.6}$$

where $a = v \pm c$. A sufficient condition for stability using a regularly spaced grid, is that all solutions of the form

$$w_j^n = \mu^n \mathrm{e}^{\mathrm{i}jk\Delta x} \tag{6.7}$$

of the Box Scheme applied to (6.6),

$$\frac{w_{j+1}^{n+1} - w_{j+1}^n}{2} + \frac{w_j^{n+1} - w_j^n}{2} + \frac{\Delta t}{\Delta x}\theta a(w_{j+1}^{n+1} - w_j^{n+1}) + \frac{\Delta t}{\Delta x}(1-\theta)a(w_{j+1}^n - w_j^n) = 0, \tag{6.8}$$

must satisfy

$$|\mu| \leq 1 \quad \forall k. \tag{6.9}$$

Substituting (6.7) into (6.8) and defining $\nu = a\dfrac{\Delta t}{\Delta x}$ yields

$$\mu = \frac{\cos\frac{k\Delta x}{2} - 2\mathrm{i}\nu(1-\theta)\sin\frac{k\Delta x}{2}}{\cos\frac{k\Delta x}{2} + 2\mathrm{i}\nu\theta\sin\frac{k\Delta x}{2}}. \tag{6.10}$$

Therefore condition (6.9) is satisfied for

$$\theta \geq \frac{1}{2}, \tag{6.11}$$

and the scheme is *unconditionally stable*. For $\theta = \frac{1}{2}$ or $\nu = 0$, we have $|\mu| = 1$ and the scheme is neutrally stable and non-dissipative (see Johnson [9]). For $\nu = 0$ there is very little dissipation in the scheme and therefore any oscillations are not damped. Hence, oscillations are likely to occur for the unsteady critical case.

Note, that for $k\Delta x = \pi$, we get

$$\mu = -\frac{1-\theta}{\theta}. \tag{6.12}$$

Using (6.7), this leads to $(-1)^n$ oscillations in the solution for $\theta = \frac{1}{2}$. For $\Delta t \to \infty$, i.e. for large ν, we also get $\mu \to -1$, which explains the grid scale oscillation we observed in the previous section.

We have shown, that the Box Scheme is unconditionally stable for the *linearised* St Venant equations, the only constraint on the time-step is due to the convergence theory of Newton's Method, which we will describe later.

Now, by the fundamental theorem of numerical methods for linear differential equations, (stated for example in [14]), consistency (showed in the previous section) and stability ensure convergence of the Box Scheme.

6.3 Stability of the Boundary Conditions

We consider the the stability of boundary conditions for the system

$$\mathbf{u}_t + f(\mathbf{u})_x = \mathbf{s}(\mathbf{u}, x). \tag{6.13}$$

We assume that the solution is known up to time level t^n and use (3.7) in order to calculate the solution at the new time level t^{n+1}. Omitting the indices for the time, we get a recurrence relation at the new time level of the form

$$\frac{\mathbf{u}_{j+1} + \mathbf{u}_j}{2} + \lambda_j \theta (\mathbf{f}_{j+1} - \mathbf{f}_j) - \frac{\Delta t}{2} \theta (\mathbf{s}_{j+1} + \mathbf{s}_j) = \mathbf{P}_j, \tag{6.14}$$

where $\mathbf{P}_j$ represents all values which are known from the previous time step. Now we may linearise this system which yields

$$\frac{\mathbf{u}_{j+1} + \mathbf{u}_j}{2} + \lambda_j \theta \mathcal{A} (\mathbf{u}_{j+1} - \mathbf{u}_j) - \frac{\Delta t}{2} \theta \mathcal{S} (\mathbf{u}_{j+1} + \mathbf{u}_j) = \Delta t \mathbf{Q}_j, \tag{6.15}$$

where $\mathbf{Q}_j$ is computed from $\mathbf{P}_j$ and further nonlinear terms. Reordering the equations gives

$$(I + 2\lambda_j \theta \mathcal{A} - \Delta t \theta \mathcal{S})\mathbf{u}_{j+1} + (I - 2\lambda_j \theta \mathcal{A} - \Delta t \theta \mathcal{S})\mathbf{u}_j = \Delta t \mathbf{Q}_j. \tag{6.16}$$

For convenience, we omit the source term $\mathcal{S}$, introduce characteristic variables $\mathbf{w} = V^{-1}\mathbf{u}$, and write $\mathbf{R}_j = V^{-1}\mathbf{Q}_j$. Then we get

$$(I + 2\lambda_j \theta \mathcal{D})\mathbf{w}_{j+1} + (I - 2\lambda_j \theta \mathcal{D})\mathbf{w}_j = \Delta t \mathbf{R}_j, \quad j = 0, \ldots, N-1 \tag{6.17}$$

where $\mathcal{D}$ is a diagonal matrix containing the eigenvalues of the Jacobian $\mathcal{A}$. If we define

$$G_j = (I + 2\lambda_j \theta \mathcal{D})^{-1}(I - 2\lambda_j \theta \mathcal{D}), \tag{6.18}$$

as an amplification matrix, then, using (6.17) the solution at time t^{n+1} may be written as a recurrence relation

$$\mathbf{w}_N = (-1)^N \left(\prod_{j=0}^{N-1} G_j \right) \mathbf{w}_0 + \Delta t \mathbf{T}_N, \tag{6.19}$$

where $\mathbf{T}_N$ is chosen appropriately. We can calculate the amplification matrix G_j which is given by

$$G_j = \begin{bmatrix} \frac{1-2\lambda_j\theta(v_j-c_j)}{1+2\lambda_j\theta(v_j-c_j)} & 0 \\ 0 & \frac{1-2\lambda_j\theta(v_j+c_j)}{1+2\lambda_j\theta(v_j+c_j)} \end{bmatrix}. \tag{6.20}$$

For supercritical flow, $v_j + c_j > v_j - c_j > 0$, both the diagonal terms have modulus less than one. So data $\mathbf{w}_0$ is transmitted in a stable manner across the domain. But, for subcritical flow, the eigenvalues of the Jacobian, $v_j - c_j < 0$ and $v_j + c_j > 0$ have opposite sign. Therefore the first component would be amplified, if it was imposed on the left. So we have to impose one boundary condition at inflow and one at outflow, in order to damp small perturbations to boundary values. Then the scheme is stable with respect to such boundary conditions.

We can also see, why problems arise, when the CFL number is very small, i.e. when one of the eigenvalues of the Jacobian $\mathcal{A}$ passes through zero. If $v_j = c_j$, then the first entry of the amplification matrix (6.20) equals unity. Using (6.19) we will find that any small perturbations to the left or right hand boundary conditions will result in perturbations in the first variable with

$$w_j^1 = (-1)^j w_0^1. \tag{6.21}$$

Also, if the CFL number is close to zero, the stability analysis in the previous section (see equation (6.10)) shows, that there is very little damping for those oscillations. This is characteristic for the so-called *washboard effect* [9], i.e. spurious period $2\Delta x$ oscillations, which we can see in Figures 15, 17 and 18. The situation gets worse for problems involving discontinuities, as it can be seen in Figure 21. Smoothing methods in order to suppress those numerical artifacts, are described in Johnson's thesis [9]. Morton and Burgess [23] investigate the stability of boundary conditions further for a different problem, but this is beyond our project.

6.4 Stability of the Thomas Algorithm

The Box Scheme gives a recurrence relation at the new time level. We want to have a closer look at the natural sweep direction in the case of a system of equations, since there are boundary conditions given on the left as well as on the right. We want to check if it is necessary to be careful about the way in which the two sweeps, in order to solve the tridiagonal system, are carried out.

The use of the recurrence relation (3.28) to calculate the values of $\mathbf{W}_j$ may be numerically unstable and lead to increasing errors, if (3.31) is not satisfied. Therefore we study E_j for the St Venant equations, in the way it has been done for the wave equation in Morton's paper[20].

For our calculations, we have to assume that A and Q are constant, such that $c_j = c$ and $v_j = v$. This is just true for the first time step (since we take constant initial conditions), but since oscillations are observed for the first time step, this assumption is suitable. For convenience we also use a regular grid, i.e. $\lambda_j = \lambda$. Then, the special form of $D = D_j$, $A = A_j$ and $C = C_j$ used in (3.22) and (3.28) gives, after some manipulations

$$E_j = \begin{bmatrix} e_j & \kappa e_j \\ \frac{1}{\kappa}(1+e_j) & 1+e_j \end{bmatrix}, \tag{6.22}$$

where $\kappa = 2\theta\lambda$ and e_j is given by the recurrence

$$e_j = -\frac{\frac{1}{2}[\kappa^2(c^2-v^2)+2v\kappa]e_{j-1} - \frac{1}{2}}{[\kappa^2(c^2-v^2)+2v\kappa]e_{j-1} - \frac{\kappa^2}{2}(c^2-v^2) + v\kappa - \frac{1}{2}}. \tag{6.23}$$

If $\kappa^2(c^2-v^2)+2v\kappa = 0$ then

$$e_j = -\frac{1}{1+2\kappa^2(c^2-v^2)}. \tag{6.24}$$

Otherwise the recurrence has two fixed points,

$$e = -\frac{v\kappa - \frac{1}{4}}{\kappa^2(c^2-v^2)+2v\kappa} \pm \sqrt{\frac{(v\kappa+\frac{1}{4})^2 + \frac{1}{2}\kappa^2(c^2-v^2)}{(\kappa^2(c^2-v^2)+2v\kappa)^2}}. \tag{6.25}$$

We can check that only the negative root is attractive and therefore

$$E_j \to \begin{bmatrix} \phi & \kappa\phi \\ \frac{1}{\kappa}(1+\phi) & 1+\phi \end{bmatrix}, \tag{6.26}$$

where

$$\phi = \frac{\frac{1}{4} - v\kappa - \sqrt{(v\kappa+\frac{1}{4})^2 + \frac{1}{2}\kappa^2(c^2-v^2)}}{\kappa^2(c^2-v^2)+2v\kappa}. \tag{6.27}$$

If the flow is transcritical, then $v = c$ and it is easy to check that $\phi = -1$ and $\|E_j\|_\infty = 1+\kappa$ as $j \to \infty$ and condition (3.31) is not satisfied for any values of λ and θ. Therefore, using the Thomas Algorithm, the solution procedure is ill-conditioned and oscillations will be amplified. In fact, (3.31) is not even satisfied near transcritical flow. For subcritical flow in contrast condition (3.31) is valid. The same results are obtained by sweeping from the left to the right first or by interchanging the order of the equation pairs A and Q, as it was done in [20].

6.5 Time-step Constraint

As we have observed in the previous section, Newton's Method only converges, if there is a restriction on the time step Δt. In order to explain this behaviour, we use the *Newton-Kantorovich Theorem*, stated in [2].

Theorem 1 (Newton-Kantorovich) *Let* $\mathbf{x}_0 \in D \subset \mathbb{R}^n$, $\mathbf{F} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ *and assume that* $\mathbf{F}$ *is continuously differentiable in* D*. Furthermore assume that the Jacobian* J *is Lipschitz continuous, i.e.*

$$\|J(\mathbf{x}) - J(\mathbf{y})\| \leq \gamma \|\mathbf{x} - \mathbf{y}\|, \forall\, \mathbf{x}, \mathbf{y} \in D, \tag{6.28}$$

with $J(\mathbf{x}_0)$ *non-singular and that there exist constants* $\beta, \eta \geq 0$ *such that*

$$\|J(\mathbf{x}_0)^{-1}\| \leq \beta, \quad \|J(\mathbf{x}_0)^{-1}\mathbf{F}(\mathbf{x}_0)\| \leq \eta, \tag{6.29}$$

with

$$\alpha := \beta\gamma\eta < \frac{1}{2}. \tag{6.30}$$

Furthermore

$$S := \{\mathbf{x} \in \mathcal{R}^n :\ \|\mathbf{x} - \mathbf{x}_0\| < t^*\} \subset D, \tag{6.31}$$

where

$$t^* = \frac{1}{\beta\gamma}(1 - \sqrt{1 - 2\alpha}). \tag{6.32}$$

Then the sequence

$$\mathbf{x}_{k+1} = \mathbf{x}_k - J(\mathbf{x}_k)^{-1}\mathbf{F}(\mathbf{x}_k), \quad k = 0, 1, \ldots \tag{6.33}$$

is well defined and converges to a unique zero of $\mathbf{F}$ *in* D*.*

If the Jacobian J is non-singular at $\mathbf{x}_0$, the constants β and η may be determined. Since the Jacobian depends on θ and λ (from (3.14) and (3.15)), detailed analysis shows that the Lipschitz constant is $\gamma = f(\theta\lambda)$, for some monotone increasing function f. We have to satisfy constraint (6.30) for convergence, i.e.

$$\eta\beta f(\theta\lambda) < \frac{1}{2}. \tag{6.34}$$

Also, the initial guess $\mathbf{x}_0$ has to be close to the solution, from (6.29) and (6.32). This is why we did not get any convergence for too large Δt in the previous model cases.

Another issue is that of transcritical flow. We have seen, that small oscillations occur near transcritical and even worse ones at supercritical flow. Therefore the stopping criterion for the iteration (3.21),

$$\frac{\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_1}{\|\mathbf{u}^{(k)}\|_1} < \text{tol}, \tag{6.35}$$

cannot be satisfied. Newton's Method converges more slowly. This is the reason, why we observe more iterations, when the flow is close to being transcritical.

7 Extension of the Box Scheme to Transcritical Flow

In this section we will describe, how to overcome the limitation of the Box Scheme using the results of Morton et al. [25] for the steady Euler equations. First we summarise the problems that arise, when the Box Scheme is applied to transcritical flow.

7.1 Invalidity of the Box Scheme for Transcritical Flow

We have discussed the Box Scheme for non-critical flow and seen some computational examples for its invalidity for transcritical flow. In [17] we will find some explanation from an engineering view point, why the Box Scheme in its usual implementation is unsuitable to model transcritical flow.

The first problem is that as the critical limit is approached, one of the eigenvalues becomes small and spurious Fourier modes are allowed to propagate uninhibited. The Preissmann Scheme becomes marginally stable if critical flow is encountered. Hence, any error (e.g. arbitrary initial conditions) will not be damped. In other words the unconditional stability of the Box Scheme will no longer be valid. This leads to a highly oscillatory solution as we have seen in section 6.

Another problem we have with transcritical flow is a counting problem. For purely subcritical or supercritical flow there are exactly two boundary conditions required and the Preissmann scheme can be directly applied. However, for transcritical flow the number of boundary conditions may differ from two. Suppose that flow in a channel starts subcritical at the upstream boundary and at some point the flow becomes supercritical. Then, to ensure well-posedness, exactly one boundary conditions is required, to be imposed at the upstream boundary. However, the Box Scheme needs two boundary conditions as stated in section 3. More examples are illustrated in [17] and [18]. There may be either too few or too many boundary conditions and the problem will be ill-posed.

A third issue is that if the boundary conditions do not correspond to the type of flow present, the solution becomes unstable. For transcritical flow it is obviously not possible to choose boundary conditions which are suitable for both subcritical and supercritical flow. This is because we are unable to ensure that each of the residuals (3.8) is zero. For each cell residual $\mathbf{R}_{j+\frac{1}{2}}$ we have to calculate two unknowns, A_j and Q_j. For subcritical flow the eigenvalues of the flux Jacobian $\mathcal{A}$ have opposite sign,

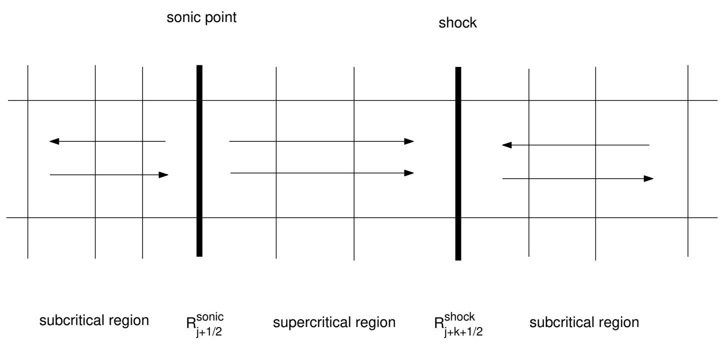

Figure 23: *Cell residuals and transcritical flow*

hence the characteristics are ingoing and outgoing, and therefore it is possible to set all the cell residuals to zero by the double sweep algorithm described above. Now, at a sonic point one eigenvalue of the Jacobian matrix $\mathcal{A}$ passes through zero and the flow becomes supercritical, i.e. both eigenvalues of the Jacobian have the same sign. This situation is illustrated in Figure 23. The cell $j + \frac{1}{2}$ contains a sonic point. The result is a *locally underdeterminded system*, since there is just one ingoing characteristic for this cell. There are not enough equations to determine the nodal value $\mathbf{u}_{j+1}$ to the right. Therefore the cell residual has to be split there. At supercritical flow both characteristics point downstream and it is possible to set both cell residuals to zero, using a single sweep method. When the flow becomes subcritical again, which corresponds to the existence of a shock in cell $j + k + \frac{1}{2}$, another problem occurs. One eigenvalue of the flux Jacobian $\mathcal{A}$ passes through zero again, and characteristics switch back to pointing one upstream and one downstream. At the shock the system is now *locally overdetermined* because of three ingoing characteristics. Therefore the cell residuals have to be combined in some way in this case.

7.2 Cell and Nodal Residuals

We have seen, that both residuals will be set to zero if the flow is wholly subcritical by using the double sweep algorithm described in [25]. From the previous section it also follows, that for wholly supercritical flow both residuals will be set to zero on the sweep from inflow to outflow. A single sweep algorithm is sufficient in this case, but the algorithm structure depends upon the flow direction. We use flow from left to right, further details on different flow directions may be found in [1]. We need to switch between those two cases for the transcritical flow.

7.2.1 Re-formulation of the Problem

To understand the mixed type flows we introduce a mapping between the cell residuals and nodal unknowns and the notation of *distribution matrices*, as it has been done in Morton et al. [25]: For each cell we can define matrices $D^{\pm}_{j-\frac{1}{2}}$. Then we can think of our Box Scheme implementation (3.8) in terms of these distributions matrices. At each interior point, we apply Newton's Method to

$$\mathbf{N}_j = D^{+}_{j-\frac{1}{2}}\mathbf{R}_{j-\frac{1}{2}} + D^{-}_{j+\frac{1}{2}}\mathbf{R}_{j+\frac{1}{2}} + \mathbf{B}_j = 0, \tag{7.1}$$

where $\mathbf{B}_j$ accounts for the boundary conditions. With equation (7.1), we have indroduced a mapping between the cell residuals and the nodal unknowns . If we define $D^{+}_{-\frac{1}{2}}$ and $D^{-}_{N+\frac{1}{2}}$ to be zero, then equation (7.1) is valid for all $N+1$ nodes $j = 0, \ldots, N$. There are certain requirements for the distribution matrices in order to set up a well-posed system which are stated in Morton et. al [25]. Now, to demonstrate this approach, we first consider the case of subcritical flow, for which discharge is imposed upstream and height is imposed downstream. The distribution matrices are then

$$D^{-}_{j+\frac{1}{2}} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad D^{+}_{j-\frac{1}{2}} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{7.2}$$

according to the sweep direction. For supercritical flow these matrices become

$$D^{-}_{j+\frac{1}{2}} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad D^{+}_{j-\frac{1}{2}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{7.3}$$

Note, that Morton et al. [24] use distribution matrices which are interchanged from those used here. The reason is that we use Newton's Method whereas Morton et al.[24] use a general update algorithm. The choice of our distribution matrices ties up with our ordering of the matrix used for the Thomas Algorithm (3.15): We start with calculating E_0 from the boundary condition plus the mass residual from its right. If we reordered our equation in (3.15), we would get the same distribution matrices as in [24].

Clearly, there is a problem in the switch from subcritical to supercritical flow, since the condition

$$\text{rank}(D^-_{j+\frac{1}{2}} + D^+_{j-\frac{1}{2}}) = 2, \tag{7.4}$$

stated in [25], is not satisfied, if a transcritical expansion fan occurs. We will describe an algorithm, by which this problem can be overcome.

The nodal residual mapping (7.1) has been used first by Morton et al. [24] and later in Johnson et al. [10]. It follows, that the matrix D^- represents the component of the cell residual to be distributed to the node to its left and D^+ represents the component of the cell residual to be distributed to the node to its right. For subcritical flow with depth D_R (or cross-sectional area A_R) imposed downstream and discharge Q_L imposed upstream, the coefficients in (7.1) become

$$\mathbf{B}_0 = \begin{bmatrix} 0 \\ Q_0^{n+1} - Q_L \end{bmatrix}, \quad \mathbf{B}_N = \begin{bmatrix} A_N^{n+1} - A_R \\ 0 \end{bmatrix}, \tag{7.5}$$

and distribution matrices as in (7.2). For supercritical flow with discharge Q_L and cross-sectional area A_L both imposed upstream we have

$$\mathbf{B}_0 = \begin{bmatrix} A_0^{n+1} - A_L \\ Q_0^{n+1} - Q_L \end{bmatrix}, \quad \mathbf{B}_N = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{7.6}$$

and distribution matrices as in (7.3).

We introduce a more sophisticated alternative to calculate the distribution matrices in (7.1) by proposing to choose the distribution matrices as a natural generalisation of those used in upwinding schemes. This is equivalent to applying a local characteristic decomposition and then assigning the residual components accordingly. To this end the Jacobian matrix $\mathcal{A}$ is decomposed as $\mathcal{A} = V\Lambda V^{-1}$ where V^{-1} denotes the row matrix of left eigenvectors and Λ is the matrix containing the eigenvalues of $\mathcal{A}$. Then the upwinding is applied to the diagonalised system. By splitting into left- and right-moving wave components, we see that this approach leads to the upwind distribution matrices

$$D^{-}_{j+\frac{1}{2}} = \tilde{V}_{j+\frac{1}{2}} \mathrm{diag}\left\{\frac{1}{2} - \frac{1}{2}\mathrm{sign}(\tilde{a}^{(k)}_{j+\frac{1}{2}}) : k = 1,2\right\} \tilde{V}^{-1}_{j+\frac{1}{2}}, \tag{7.7}$$

$$D^{+}_{j-\frac{1}{2}} = \tilde{V}_{j-\frac{1}{2}} \mathrm{diag}\left\{\frac{1}{2} + \frac{1}{2}\mathrm{sign}(\tilde{a}^{(k)}_{j-\frac{1}{2}}) : k = 1,2\right\} \tilde{V}^{-1}_{j-\frac{1}{2}}, \tag{7.8}$$

where $\tilde{\mathcal{A}}_{j\pm\frac{1}{2}}$ is an average value of $\mathcal{A}$ for the cell and $\tilde{a}^{(k)}_{j\pm\frac{1}{2}}$, $k = 1, 2$, are its eigenvalues. The best choice of $\tilde{\mathcal{A}}_{j+\frac{1}{2}}$ is the *Roe average matrix* (see [4] or [10]), which may be written as

$$\tilde{\mathcal{A}}_{j+\frac{1}{2}} = \begin{bmatrix} 0 & 1 \\ \tilde{c}^2_{j+\frac{1}{2}} - \tilde{v}^2_{j+\frac{1}{2}} & 2\tilde{v}_{j+\frac{1}{2}} \end{bmatrix}, \tag{7.9}$$

where

$$\tilde{c}_{j+\frac{1}{2}} = \sqrt{\frac{c_j^2 + c_{j+1}^2}{2}} \quad \text{and} \quad \tilde{v}_{j+\frac{1}{2}} = \frac{v_j c_j + v_{j+1} c_{j+1}}{c_j + c_{j+1}}, \quad j = 0, \ldots, N-1. \tag{7.10}$$

This choice of distribution matrices has a conservation property which is shown in Morton et al. [25]. We should also note, that this choice of the distribution matrices corresponds to the matrices used in *Roe's approximate Riemann solver*, see [13, page 142] and [22, page 53], since we use a natural generalisation of the matrices used in upwinding schemes.

7.2.2 Implementation

Suppose our flow is subcritical at inflow and outflow and there is an interior supercritical region as in problems (P3) and (P4c). Then our Box Scheme algorithm may be used in a modified version, which is a combination of that described in [25] and the double sweep Thomas Algorithm. Note, that there might be no supercritical region in the first place. Since we consider the unsteady case the supercritical interior region might occur after some time $t > t_0$.

First, we find the region of supercritical flow in the domain, by checking the Froude Number.

The Subcritical Region In the subcritical region the nodal values are given from two cell equations, one from either side. Therefore we start our algorithm as usual, using sweeping from left to right, until the left of the sonic cell is reached.

Splitting at Sonic Point Suppose the cell $j+\frac{1}{2}$, contains the sonic point x_s. At this point, we have the momentum residual equation $R^{(2)}_{j+\frac{1}{2}}$ from the left, but no equation from the right. Therefore the mass cell residual $R^{(1)}_{j+\frac{1}{2}}$ will be needed both to update the subcritical node on its left and the supercritical node on its right. Hence, this mass cell residual is split at the predicted sonic point x_s. The sonic point position is found by interpolation of the Froude Number. Finally, the subcritical partial residual $R^{(1)-}_{j+\frac{1}{2}}$ is used to update the last subcritical state vector $\mathbf{u}_j$ and the supercritical partial residual $R^{(1)+}_{j+\frac{1}{2}}$ may be taken as initial condition for the sweep through the supercritical cell. The momentum cell residual keeps unchanged.

The Supercritical Region The supercritical cells use both components of the residual to determine the nodal value on the right. We continue this to the left of the shock cell.

Treatment of the Shock At the shock cell, the nodal value on the left is already determined, but that on the right would have to set both shock cell residual components to zero, as well as the mass residual $R^{(1)}$ from the cell to the right. The node on the right of the cell is determined from the momentum residual $R^{(2)}$ of the cell and the combined mass residuals $R^{(1)}$ from the two cells either side of it. This approach satisfies the *discrete conservation law* for the box scheme. Again, we keep the momentum cell residual unchanged.

The Subcritical region Finally we can continue the double sweep through the subcritical region.

7.2.3 Solution Procedure

It is important to note, that we are using two different kinds of algorithms. For subcritical regions, where the system is block-tridiagonal, we use the efficient double sweep method as it is described in section 3.4. The easier single sweeping is used for the supercritical region, where the system is block-tridiagonal, but has only got 2×2 block entries on the diagonal and the subdiagonal.

For transcritical flow with an interior supercritical region as described above, the modified Newton matrix from (3.15) still has banded structure and may be written in the tridiagonal form (3.22). Due to the splitting of the mass residual at the sonic cell, we introduce one more equation. Assume the sonic point lies in the cell $[x_j, x_{j+1}]$. We interpolate the sonic point by either interpolating the Froude Number,

$$x_{\text{sonic}} = (1 - F_j)\frac{x_{j+1} - x_j}{F_{j+1} - F_j} + x_j, \tag{7.11}$$

or by linear interpolation of the wave speed $a = v - c$, where $a_j < 0 < a_{j+1}$,

$$x_{\text{sonic}} = -a_j\frac{x_{j+1} - x_j}{a_{j+1} - a_j} + x_j. \tag{7.12}$$

It is obvious, that both approaches give the same result. Splitting the residual and setting both parts to zero,

$$R^{(1)-}_{j+\frac{1}{2}} = 0 \quad \text{and} \quad R^{(1)+}_{j+\frac{1}{2}} = 0, \tag{7.13}$$

is consistent with taking the derivative of the equilibrium equation (see Morton et al. [25, page 223]). Hence, the solution is smoothed at the sonic point and the oscillations observed in section 5 should vanish. The following matrix shows the structure of the matrix used for the Thomas Algorithm at the interface from subcritical to supercritical flow, i.e. at the sonic point.

$$
M_1 = \begin{array}{cc}
\left[\begin{array}{cc|cc|cc|cc|c}
x & x & & & & & & & \\
x & x & x & x & & & & & \\
\hline
x & x & x & x & & & & & \\
 & & x^- & x^- & x^- & x^- & & & \\
\hline
 & & x^+ & x^+ & x^+ & x^+ & & & \\
 & & x & x & x & x & & & \\
\hline
 & & & & x & x & x & x & \\
 & & & & x & x & x & x & \\
\hline
 & & & & & & & & \dots
\end{array}\right]
&
\begin{array}{c}
\\
R^{(1)} \\
R^{(2)} \\
R^{(1)-} \\
R^{(1)+} \\
R^{(2)} \\
R^{(1)} \\
R^{(2)} \\
\\
\end{array}
\end{array}
\tag{7.14}
$$

From (7.14) we see, that C_j, the superdiagonal block entries, become zero in the supercritical region, due to the residual splitting at the sonic point.

At the shock we have to combine two cells, since we have a locally overdetermined system. In order to find out, which cells we need to combine, we identify a cell as subcritical or supercritical by averaging the Froude Number. Assume we have found that the node $\mathbf{u}_j$ is supercritical and the node $\mathbf{u}_{j+1}$ is subcritical. We need to find out if the cell is sub- or supercritical. To this end we calculate the Froude Number for the cell $j+\frac{1}{2}$ by

$$F_{j+\frac{1}{2}} = \frac{1}{2}(F_j + F_{j+1}). \tag{7.15}$$

If $F_{j+\frac{1}{2}} > 1$ we set the cell to be supercritical and we combine cells $j+\frac{1}{2}$ and $j+\frac{3}{2}$ by

$$R^{(1)}_{j+\frac{1}{2}} + R^{(1)}_{j+\frac{3}{2}} = 0. \tag{7.16}$$

Similarly, if $F_{j+\frac{1}{2}} < 1$ we set the cell to be subcritical and we have to combine the cells $j-\frac{1}{2}$ and $j+\frac{1}{2}$ by

$$R^{(1)}_{j-\frac{1}{2}} + R^{(1)}_{j+\frac{1}{2}} = 0. \tag{7.17}$$

Another possibility to determine, if a cell is sub- or supercritical, is to calculate the Roe matrix (7.9) for this cell and to check the sign of its eigenvalues. The following matrix shows the structure of the Newton matrix at the interface from supercritical to subcritical flow, i.e. at the shock.

$$M_2 = \left[\begin{array}{cc|cc|cc|cc|cc}
\cdots & & & & & & & & & \\ \hline
x & x & & & & & & & & \\
x & x & & & & & & & & \\ \hline
x & x & x & x & & & & & & \\
x & x & x & x & & & & & & \\ \hline
 & & x_1 & x_1 & x_1+x_2 & x_1+x_2 & x_2 & x_2 & & \\
 & & x & x & x & x & & & & \\ \hline
 & & & & x & x & x & x & & \\
 & & & & & & x & x & x & x \\ \hline
 & & & & & & x & x & x & x \\
 & & & & & & & & x & x \\ \hline
 & & & & & & & & & \cdots
\end{array}\right]
\quad
\begin{array}{c}
\\
R^{(1)} \\
R^{(2)} \\
R^{(1)} \\
R^{(2)} \\
R_1^{(1)} + R_2^{(1)} \\
R^{(2)} \\
R^{(2)} \\
R^{(1)} \\
R^{(2)} \\
R^{(1)} \\
\\
\end{array} \tag{7.18}$$

From (7.18) we see, that C_j, the superdiagonal block entries, only equal zero for the supercritical region. Due to the combination of the cell residuals at the shock cell we have $C_j \neq 0$ for the subcritical flow region.

We sweep from left to right and calculate E_j and $\mathbf{F}_j$ successively by the forward recursions (3.29) and (3.30). E_0 is calculated from the boundary condition plus the first mass residual $R_{\frac{1}{2}}^{(1)}$ from its right. For the supercritical region C_j and therefore E_j equals zero, and we can omit the forward recursion. We sweep back in order to calculate $\mathbf{W}_j$ by (3.27) and (3.28). Since $E_j = 0$ for the supercritical region we only have to compute a back substitution in this case.

7.2.4 Overall Algorithm Description

We now summarise the overall algorithm. The setup is done as in the subcritical case (see section 3.5). Then, for each time step we perform the following algorithm, until the steady state is reached.

1. check if there is a supercritical region, i.e. if $F > 1$. If so, go to 2., otherwise

use the unmodified Thomas Algorithm, described in section 3.5;

2. find the region of supercritical flow, i.e. the indices j with $F_j > 1$;

3. find the location of the sonic point by interpolation using (7.11) or (7.12);

4. find the cell with $F_j > 1$ at the node to its left and $F_{j+1} < 1$ at the node to its right, i.e. the shock cell and determine the shock cell to be subcritical or supercritical by (7.15);

5. set up the A_j, C_j, D_j and $\mathbf{Z}_j$ for all $j = 0, \ldots, N$ similar to the subcritical case (see section 3.5), but split the residuals at the sonic cell using (7.13) and combine the residuals at the shock cell using (7.16) or (7.17);

6. calculate E_j and $\mathbf{F}_j$ by the forward recursions (3.29) and (3.30), note that $E_j = 0$ for the supercritical region;

7. solve the system by the Thomas Algorithm (3.27) and (3.28);

8. calculate the new iterate using (3.20);

9. check, if stopping criterion (3.21) is satisfied. If so, then move on to the next time level, otherwise start with 1. again.

Since we use Newton's Method, the number of iterations for each time step should be small.

7.3 Numerical Results

We consider the transcritical problems (P3) and (P4c) of section 5, again and apply them to our modified Box Scheme. Clearly, our modified scheme still works for the subcritical and near-critical problems (P1), (P2), (P4a) and (P4b).

7.3.1 Transcritical Problem involving a Hydraulic Jump (P3)

We recall problem (P3) and take exactly the same setup as described in section 5. The scheme broke down after $t = 51$.

We also want to compare our results with the scheme used by Johnson et al. [10]. They propose to overcome the lack of equations that arises at the sonic point, i.e.

$$\mathrm{rank}(D^-_{j+\frac{1}{2}} + D^+_{j-\frac{1}{2}}) \leq 1, \tag{7.19}$$

by introducing an extra linearly independent equation. They suggest to impose a so-called internal boundary condition $\mathbf{B}_j$,

$$\mathbf{B}_j = (I - D^-_{j+\frac{1}{2}} - D^+_{j-\frac{1}{2}})\Delta\mathbf{u}_j, \tag{7.20}$$

which clearly provides the missing extra linearly independent equation. They also show, that the choice of this boundary condition does not contradict the discrete conservation law of the original Box Scheme. If we implement this scheme for our problem (P3) we obtain the steady state solution in Figure 24. Oscillations occur

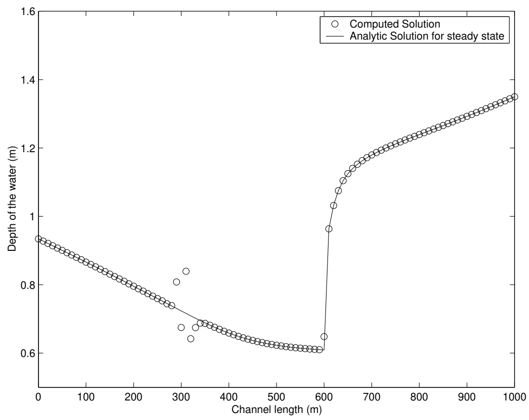

Figure 24: *Transcritical Problem $(P3)$ with internal boundary conditions*

at the sonic point, which are clearly not right and which we want to avoid. Furthermore, the convergence is very slow, as soon as the Froude Number approaches unity. We want to avoid that.

Therefore, we test our modified Box Scheme algorithm which satisfies the discrete conservation law and compare it to Johnson's results. We use a time step of $\Delta t = 1$. The maximum Froude Number for the initial state is 0.3749 again, and it increases with time. We have just about $3 - 5$ iterations for convergence at each time step, even when the Froude Number is equal to or larger than one. Period $2\Delta x$ oscillations occur in the supercritical region. But it is important to note, that the scheme does not break down this time. The reasons for the oscillations, which may be seen in Figures 25 and 26 have been explained in section 6: We can switch between the subcritical and supercritical flow regions, but the Froude Number has to pass unity, which entails that one of the eigenvalues of the Jacobian passes through zero and the CFL number is very small.

In Figure 27 we can see the analytic and numeric solution for the depth of the water at steady state, which is indeed very adequate and better than Johnson's solution. We can also see that any oscillations have died out. The maximum Froude Number for that state is 1.2988 and the maximum CFL number is $\nu_{max} = 0.7102$. A plot of the Froude Numbers for the steady state can be seen in Figure 28. We should also note,

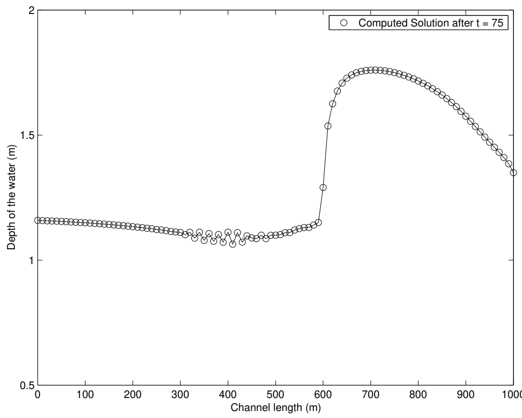

Figure 25: *Transcritical problem approaching Froude Number $F = 1$ after time $t = 75$ $(P3)$*

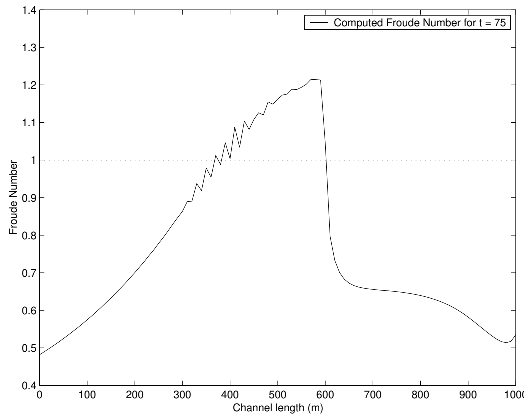

Figure 26: *Approximation to Froude Number for transcritical problem after time $t = 75$ $(P3)$*

that we cannot choose the time step Δt too large. As long as the flow is subcritical large time steps are fine, but as the critical region with Froude Number $F \approx 1$ is approached with large Δt, occurring oscillations are too large, so that Newton's Method fails to converge. This problem suggests the use of adaptive time steps, according to the size of the Froude Number.

7.3.2 Transcritical Problem with Changing Channel Width (P4c)

We recall problem (P4c) and take exactly the same setup as described in section 5. The scheme broke down for our original Box Scheme implementation. We use a

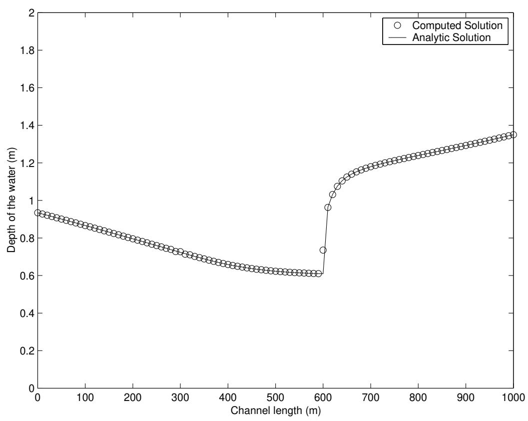

Figure 27: *Transcritical problem at steady state* $(P3)$

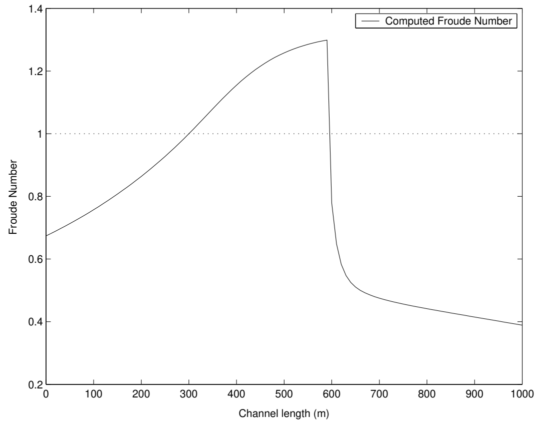

Figure 28: *Approximation to Froude Number at steady state* $(P3)$

time step of $\Delta t = 2$ for our modified code. The maximum Froude Number for the initial state is 0.8616 again, and it increases with time. We have just about $2-5$ iterations for convergence at each time step, even when the Froude Number approaches unity. Period $2\Delta x$ oscillations occur in the supercritical region, but the scheme does not break down this time. The reasons for the oscillations, which may be seen in Figure 29 have been explained in section 6: We can switch between the subcritical and supercritical flow regions, but the Froude Number passes unity, which entails that one of the eigenvalues of the Jacobian passes through zero and the CFL number is very small.

In Figure 29 we can see the numeric solution for the steady state. There are still oscillations at the steady state, due to the discontinuity of the solution. The maximum Froude Number for that state is 2.0616 and the maximum CFL number is $\nu_{max} = 0.1947$. As in the previous problem we cannot choose the time step Δt too large. As long as the flow is subcritical large time steps are fine, but as the critical region with Froude Number $F \approx 1$ is approached with a large value of Δt, occurring oscillations are too large, so that Newton's Method fails to converge. This problem suggests the use of adaptive time steps, according to the size of the Froude Number. Figure 30 shows approximations to the Froude number for the steady state.

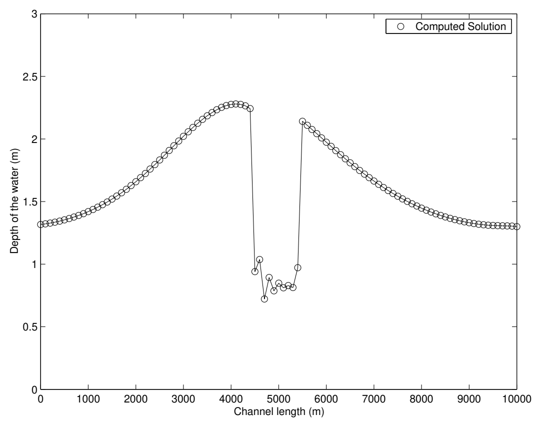

Figure 29: *Transcritical problem at steady state* $(P4c)$

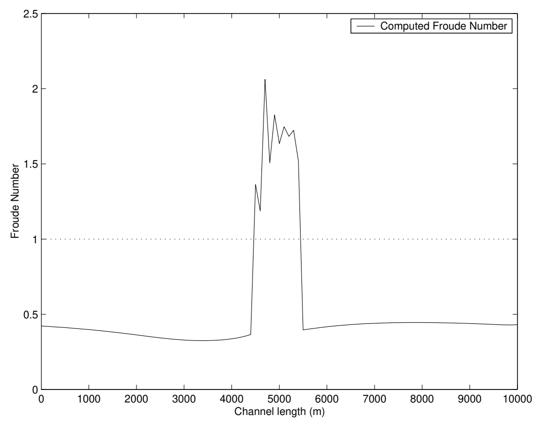

Figure 30: *Approximation to Froude Number at steady state* $(P4c)$

7.4 Local Post Processing and Shock Fitting

The Box Scheme is inadequate for computing discontinuous solutions of nonlinear conservation laws as stated and explained in Mitchell's thesis [19]. We consider again the numerical results for the depth D and the discharge Q for the application of our *modified Box Scheme* to the transcritical problem (P3). The results for the steady state are shown in Figure 31. Clearly the isolated point at the shock does not satisfy

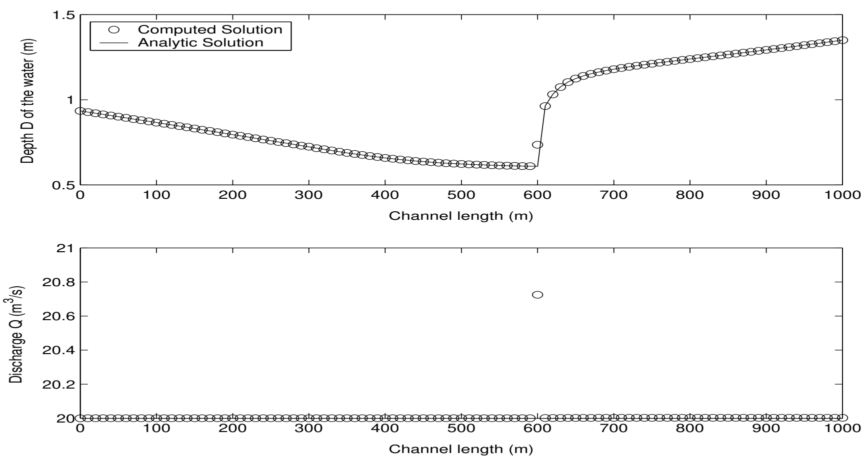

Figure 31: *Depth D and discharge Q at steady state* $(P3)$

the mass conservation law and therefore is incorrect. We can explain the position of the isolated point in the discharge Q by selecting a variable that has an intermediate value, e.g. if the velocity should have a viscous term in its equation and we compute the limit as it goes to zero, the velocity should have an intermediate value. If we compute the velocity $v = \frac{Q}{A}$ at each point, we will see that the velocity has indeed got an intermediate value at the shock.

In order to suppress the isolated point, we use *shock fitting* and introduce a discontinuity in the shock cell. It is for this reason we look closer at the shock region, see Figure 32. Suppose the shock is between the nodes $k-1$ and $k+1$ with the nodal value Q_k out of line. We assume that all other values for the depth D_j and the discharge Q_j, $j = 1, \ldots, k-1, k+1, \ldots, N$, are computed correctly.

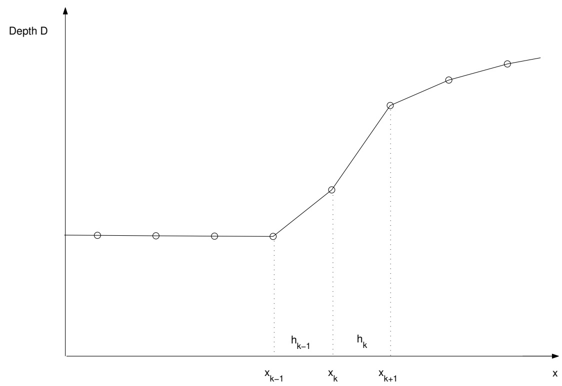

Figure 32: *Diagram showing depth function at the shock before introducing a discontinuity*

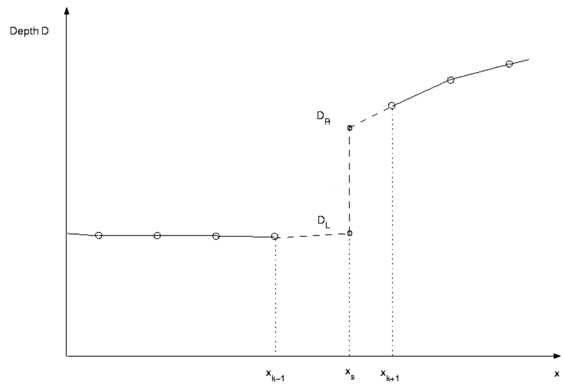

Figure 33: *Diagram showing depth function and shock location x_s after introducing a discontinuity*

We use equations (2.33)-(2.36), which simplify for the steady state, since $\mathbf{u}_t = 0$. Then we apply mass conservation, which implies $Q_x = 0$ for the steady state, over the cells $[x_{k-1}, x_k]$ and $[x_k, x_{k+1}]$, i.e.

$$\frac{Q_k - Q_{k-1}}{\Delta x} = 0 \quad \text{and} \quad \frac{Q_{k+1} - Q_k}{\Delta x} = 0. \tag{7.21}$$

This approach gives $Q_k = Q$, the constant value for the discharge in the steady case. Furthermore we consider the depth D and introduce a discontinuity at the shock x_s, see Figure 33. We call x_s the *shock position*, which is unknown. We may write

$$x_s = x_{k-1} + \theta(h_{k-1} + h_k), \tag{7.22}$$

where h_{k-1} and h_k are the sizes of the cells left and right of the shock (see Figure 32) and $\theta \in [0, 1]$. Now, we apply momentum conservation over the cells $[x_{k-1}, x_s]$ and $[x_s, x_{k+1}]$. Using (2.33)-(2.36), and the simplifications (3.11), which are valid for our problem (P3), the momentum conservation law for the steady state is

$$\left[\frac{Q^2}{A} + g\left(\frac{D^2B}{2} + \frac{D^3S_T}{3}\right)\right]_x = gA(S_0 - S_f). \tag{7.23}$$

Applying this momentum conservation law over both the cells $[x_{k-1}, x_s]$ and $[x_s, x_{k+1}]$ we get

$$\begin{aligned} \frac{F_s^- - F_{k-1}}{\theta(h_{k-1}+h_k)} &= \frac{S_{k-1}+S_s^-}{2}, && (7.24)\\ \frac{F_{k+1} - F_s^+}{(1-\theta)(h_{k-1}+h_k)} &= \frac{S_s^+ + S_{k+1}}{2}, && (7.25)\end{aligned}$$

where

$$\begin{aligned} F_{k\pm1} &= \frac{Q^2}{A_{k\pm1}} + g\left(\frac{D_{k\pm1}^2 B}{2} + \frac{D_{k\pm1}^3 S_T}{3}\right), && (7.26)\\ S_{k\pm1} &= gA(S_0 - S_f), && (7.27)\end{aligned}$$

with S_0 and S_f evaluated at $x_{k\pm1}$ using formula (2.25). Clearly, the values of $F_{k\pm1}$ and $S_{k\pm1}$ are known from the assumption that all values except the ones at the node x_k are computed correctly. Furthermore we have

$$F_s^- = \frac{Q^2}{A_L} + g\left(\frac{D_L^2B}{2} + \frac{D_L^3S_T}{3}\right) \quad \text{and} \quad F_s^+ = \frac{Q^2}{A_R} + g\left(\frac{D_R^2B}{2} + \frac{D_R^3S_T}{3}\right), \tag{7.28}$$

where D_L (A_L, respectively) is the depth (cross-sectional area, respectively) left of the shock and D_R (A_R, respectively) is the depth (cross-sectional area, respectively) right of the shock, as can be seen in Figure 33. With equations (7.24) and (7.25) for the momentum conservation over the two cells, we have got 2 equations to satisfy for our problem. But with θ, D_L and D_R (or A_L and A_R respectively, since depth and cross-sectional area can be determined from each other, see equation (2.31)) we get 3 unknowns. Therefore we need one more equation, which is given by the *Rankine-Hugoniot jump condition* (see [13],[12]) for the shock speed. It is the speed at which a discontinuity must move in order to be a weak solution [22]. For scalar problems, it is simply

$$s = \frac{f(u_L) - f(u_R)}{u_L - u_R} = \frac{[f]}{[u]}, \tag{7.29}$$

where u_L and u_R are the states left and right of the shock, s is the shock speed and $f(u)$ is the flux function. Since we only consider the momentum conservation law, we deal with a scalar problem. Furthermore our problem is steady, i.e. the shock does not move and therefore we have

$$s = [f] = 0. \tag{7.30}$$

Applied to our problem, the *Rankine-Hugoniot jump condition* becomes

$$\frac{Q^2}{A_L} + g\left(\frac{D_L^2 B}{2} + \frac{D_L^3 S_T}{3}\right) = \frac{Q^2}{A_R} + g\left(\frac{D_R^2 B}{2} + \frac{D_R^3 S_T}{3}\right). \tag{7.31}$$

With equations (7.24), (7.25) and (7.31), we now have 3 equations, which we may write in the form

$$\mathbf{H}(D_L, D_R, \theta) = 0, \tag{7.32}$$

for our 3 unknowns D_L, D_R and θ. We will use Newton's Method in order to solve the nonlinear system (7.32). If we write $\mathbf{y} = (D_L, D_R, \theta)^T$, then Newton's Method for the given starting guess $\mathbf{y}^0$ is

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \mathbf{d}^k, \quad \text{where} \quad \mathbf{H}_{\mathbf{y}}(\mathbf{y}^k)\mathbf{d}^k = -\mathbf{H}(\mathbf{y}^k), \quad k \geq 0, \tag{7.33}$$

as stated in [30]. As initial guesses we take $\theta^0 = \frac{1}{2}$ and the computed values left and right of the shock, i.e. $D_L^0 = D_{k-1}$ and $D_R^0 = D_{k+1}$. As stopping criterion we use

$$\|\mathbf{d}^k\|_\infty < \text{tol}, \tag{7.34}$$

where $tol = 10^{-8}$. With our initial guess, we indeed get very good results for the

	INITIAL GUESS	RESULT
D_L	0.61	0.609281
D_R	0.96	0.850282
θ	0.5	0.476177

Table 2: *Results of the shock fitting*

depths left and right of the shock D_L and D_R and the shock position θ. They may be seen in table 2. We need only 5 iterations for convergence with Newton's method, which is known to converge quadratically. In Figure 34 the results that we obtained

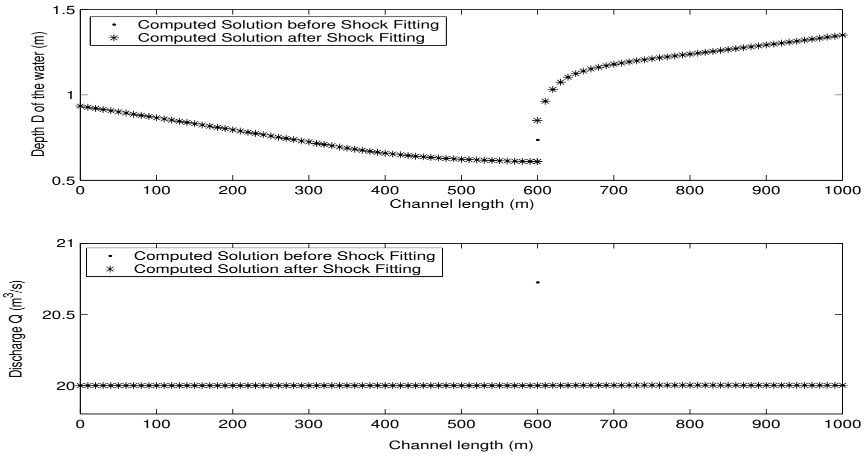

Figure 34: *Depth D and discharge Q at steady state $(P3)$ after shock fitting*

by shock fitting are compared to the ones without shock fitting. We can see that our shock recovery procedure indeed works very well. The final solution that we get for the steady state is very satisfactory and adequate. Of course we can apply the same shock fitting procedure to problem (P4c).

8 Conclusions and Future Work

In this report we have investigated the Box Scheme applied to both non-critical and transcritical flow.

We gave a detailed derivation of the unsteady St Venant equations for open channel flow and explained how to apply the Box Scheme to that system of equations. Special model problems were created, including both entirely subcritical and transcritical flow. By applying the Box Scheme to those test problems, we found that the scheme in its original implementation works very adequate for both wholly subcritical and supercritical problems, but breaks down for transcritical flow. We did extensive analysis on the method and found explanations for its breakdown in the transcritical case.

From those results we were able to modify the scheme, such that it could then be applied to transcritical flow. The modification of the Box Scheme was based on the work done by Morton et al. [25] for the case of the steady Euler equation. We applied this *modified Box Scheme* to the unsteady St Venant equations. Since we used Newton's Method in order to solve the nonlinear system that arised from the Box Scheme at each time step, we observed very fast convergence. We also found that the solution of the linear system at each Newton iteration was very efficient, since we were able to use the Thomas Algorithm for block-tridiagonal matrices, even in the case of transcritical flow. The code is fast and easily adaptable. The results that we obtained indeed were very accurate, see Figures 27 and 29. Finally, we have shown, that the method of *shock fitting* works particularly successful in the recovery of the shock discontinuity, see Figure 34.

Overall, the developed methods worked very well for both non-critical and transcritical unsteady flow.

Further work includes the introduction of more than one supercritical interior region in the problem. The analysis and implementation so far suggests that the modified Box Scheme will still work satisfactorily. We also propose to do further investigations on other nonlinear iteration techniques than Newton's Method in order to compare the results. Furthermore, it should be very interesting to see the shock fitting, that was done in section 7.4 for the steady state applied to the result after each time step, i.e. to the unsteady problem. We therefore should adapt the methods developed for the scalar case in [19] and apply them to our system of equations. In addition, we could extend our results to two-dimensional flows.

A MATLAB Code for Creating Model Problems

In section 4 we explained how to create steady state problems in order to test our algorithms. By using equation (4.3) we can calculate the bed slope function $S_0(x)$. We have to integrate this function numerically, in order to get the function for the channel bed itself. The MATLAB-code for the test problems may be found in

`/home/mamamf/Project/work/testproblems`.

We briefly describe the programs.

model1.m creates test problem (P1) using the algorithm stated in section 4.1 and visualises depth and bed slope

integrate1.m integrates the bed slope function numerically in order to the function for the height of the channel bed find

model2.m creates test problem (P2) using the algorithm stated in section 4.1 and visualises depth and bed slope

integrate2.m see *integrate1.m*

model3.m creates test problem (P3) using the algorithm stated in section 4.1 and visualises depth and bed slope

integrate3.m see *integrate1.m*

critical.m function which calculates the critical depth for test problems, i.e. the depth with Froude Number $Fr = 1$

B MATLAB Code for Solving Non-critical Problems with the Box Scheme

In section 5 we found the solutions to our steady state test problems and we got indeed very good results for subcritical flows, but problems with the Box Scheme for transcritical flows. The MATLAB-code for these Box Scheme implementations may be found in

`/home/mamamf/Project/work/cell`.

We briefly describe the programs.

svcellfull.m solves the unsteady inhomogeneous St Venant equations with bed slope function given in (P1) or (P2) with Newton's Method by setting the cell residuals to zero, the Thomas Algorithm is used to solve the system, works only for subcritical problems as (P1) and (P2)

svcell.m similar to *svcellfull.m*, but omits source term in the Jacobian for Newton's Method and is therefore only a Quasi-Newton Method

svcelltransfull.m solves the unsteady inhomogeneous St Venant equations with bed slope function given in (P3) with Newton's Method by setting the cell residuals to zero, the Thomas Algorithm is used to solve the system, breaks down, as soon as problem becomes transcritical

svcelltrans.m similar to *svcelltransfull.m*, but omits source term in the Jacobian for Newton's Method and is therefore only a Quasi-Newton Method

chwidthfull.m solves the unsteady inhomogeneous St Venant equations with changing channel width and bed slope function given in (P4) with Newton's Method by setting the cell residuals to zero, the Thomas Algorithm is used to solve the system, works fine for gentle bed slope, breaks down, as soon as problem becomes transcritical for steep bed slope

chwidth.m similar to *chwidthfullfull.m*, but omits source term in the Jacobian for Newton's Method and is therefore only a Quasi-Newton Method

dambreakfull.m solves the dam-break problem (unsteady) for the (in)homogeneous St Venant equations with Newton's Method by setting the cell residuals to zero, the Thomas Algorithm is used to solve the system, if the dam height is too large the problem becomes transcritical and the computation breaks down

dambreak.m similar to *dambreakfull.m*, but omits source term in the Jacobian for Newton's Method and is therefore only a Quasi-Newton Method

getjac.m function which calculates the local Jacobian for each cell (without including the source term)

getsjac.m function which calculates the local source term Jacobian

getres.m function which calculates the local cell residual

gj.m similar to *getjac.m*, but includes the derivative of the function for the channel width

gs.m similar to *getsjac.m*, but includes the derivative of the function for the channel width

gr.m similar to *getres.m*, but includes the derivative of the function for the channel width

C MATLAB Code for Solving Transcritical Problems with the Box Scheme using Cell Residuals

In section 7.2 we found the solutions to steady state test problems with the modified Box Scheme for transcritical flows. The MATLAB-code for the modified Box Scheme may be found in

`/home/mamamf/Project/work/cell_trans`.

We briefly describe the programs.

svcelltransfull.m solves the unsteady inhomogeneous St Venant equations with bed slope function given in (P3) with Newton's Method by setting the cell residuals to zero, the modified Box Scheme algorithm which is described in section 7.2 is used to solve the system, works for transcritical flow

chwidthfull.m solves the unsteady inhomogeneous St Venant equations with changing channel width and bed slope function given in (P4) with Newton's Method by setting the cell residuals to zero, the modified Box Scheme algorithm which is described in section 7.2 is used to solve the system, works for both gentle and steep bed slope, i.e. transcritical flow

dambreakfull.m solves the dam-break problem (unsteady) for the (in)homogeneous St Venant equations with Newton's Method by setting the cell residuals to zero, the Thomas Algorithm is used to solve the system, works for transcritical flow

getjac.m see appendix B

getsjac.m see appendix B

getres.m see appendix B

gj.m see appendix B

gs.m see appendix B

gr.m see appendix B

roe.m function which calculates the Roe matrix and its eigenvalues and eigenvectors

roewidth.m similar to *roe.m*, but includes the changing channel width in the Roe matrix

D MATLAB Code for Solving Transcritical Problems with the Box Scheme using Nodal Residuals

In section 7.2 we described the solutions to steady state test problems using nodal residuals and distribution matrices. In this section we include the MATLAB-code for the nodal residuals with the "internal boundary conditions approach" used by Johnson et al. (see [9],[10]). We have seen, that this approach gives undesirable oscillations at the sonic point for transcritical flow. The code is available from

`/home/mamamf/Project/work/node`.

We briefly describe the programs.

svnodefull.m solves the unsteady inhomogeneous St Venant equations with bed slope function given in (P1) or (P2) with Newton's Method by setting the nodal residuals (7.1) to zero, the Thomas algorithm is used to solve the system, works only for subcritical problems as (P1) and (P2), this method gives exactly the same results with the same number of iterations as *svcellfull.m*, see appendix B

svnode.m similar to *svnodefull.m*, but omits source term in the Jacobian for Newton's Method and is therefore only a Quasi-Newton Method, this method gives exactly the same results with the same number of iterations as *svcell.m*, see appendix B

nodetransinternal.m solves the unsteady inhomogeneous St Venant equations with bed slope function given in (P3) with Newton's method by setting the nodal residuals (7.1) to zero, the Thomas Algorithm is used to solve the system, we use internal boundary conditions as in Johnson [9] in order to overcome the problems at the transcritical expansion fan, takes far more iterations as our modified algorithm in *svcelltransfull.m* (see appendix C) and gives oscillations at the sonic point (see Figure 24)

nodetransinternalwidth.m solves the unsteady inhomogeneous St Venant equations with bed slope function given in (P4) with Newton's method by setting the nodal residuals (7.1) to zero, the Thomas Algorithm is used to solve the system, we use internal boundary conditions as in Johnson [9] in order to overcome the problems at the transcritical expansion fan, takes far more iterations as our modified algorithm *chwidthfull.m* (see appendix C)

jac.m function which calculates the local Jacobian for each cell (without including the source term)

sjac.m function which calculates the local source term Jacobian

res.m function which calculates the local cell residual

gj.m similar to *jac.m*, but includes the derivative of the function for the channel width

gs.m similar to *sjac.m*, but includes the derivative of the function for the channel width

gr.m similar to *res.m*, but includes the derivative of the function for the channel width

meanjac.m function which calculates an average Jacobian for a cell taking the mean values of its neighbouring nodes (similar to Roe matrix), eigenvalues and eigenvectors of this Jacobian are also computed

roe.m see appendix C

roewidth.m see appendix C

E MATLAB Code for Shock Fitting

In section 7.4 we applied shock fitting to the steady state. We calculated the left and right states of the shock and the shock position by applying Newton's Method to a nonlinear system of equations. The MATLAB-code for the Newton iteration may be found in

`/home/mamamf/Project/work/shockfitting`.

We briefly describe the programs.

shockfitting.m main program, which does the set up and the actual Newton iteration (7.33) until the stopping criterion (7.34) is satisfied.

rhsfun.m function, which determines the value of $\mathbf{H}(\mathbf{y})$ at each step of the iteration.

funjacobian.m function, which determines the Jacobian $\mathbf{H}_{\mathbf{y}}(\mathbf{y})$ at each step of the iteration

References

[1] ABBOTT, M. B., AND BASCO, D. R. *Computational Fluid Dynamics - An Introduction for Engineers*. Longman Scientific and Technical, UK, 1989.

[2] DENNIS, J. E., AND SCHNABEL, R. B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall, New Jersey, 1983.

[3] FOWLER, A. C. *Mathematical Models in the Applied Sciences*. Cambridge University Press, 1997.

[4] GODLEWSKI, E., AND RAVIART, P.-A. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer Verlag New York, Inc., 1996.

[5] GOLUB, G., AND LOAN, C. V. *Matrix Computations*, 3nd ed. John Hopkins University Press, Baltimore, 1996.

[6] GROSSMANN, C., AND ROOS, H.-G. *Numerik partieller Differentialgleichungen*, 2nd ed. B.G. Teubner, Stuttgart, 1994.

[7] HENDERSON, F. M. *Open Channel Flow*. McMillan Company, New York, 1966.

[8] HIGHAM, D. J., AND HIGHAM, N. J. *MATLAB Guide*. SIAM, Philadelphia, 2000.

[9] JOHNSON, T. C. *Implicit numerical schemes for transcritical shallow water flow*. PhD thesis, Department of Mathematics, University of Reading, 2001.

[10] JOHNSON, T. C., BAINES, M. J., AND SWEBY, P. K. A box scheme for transcritical flow. *International Journal for Numerical Methods in Engineering 55* (2002), 895–912.

[11] KUTIJA, V. On the numerical modelling of supercritical flow. *Journal of Hydraulic Research 31*, 6 (1993), 841–859.

[12] LAX, P. D. *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*. Regional conference series in applied mathematics. Society for Industrial and Applied Mathematics, Philadelphia, 1973.

[13] LEVEQUE, R. J. *Numerical Methods for Conservation Laws*. Birkhäuser Verlag, Basel, 1992.

[14] LEVEQUE, R. J. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[15] MACDONALD, I. Test Problems with Analytic Solutions for Steady Open Channel flow. Numerical analysis report 6/94, University of Reading, Department of Mathematics, 1995.

[16] MACDONALD, I., BAINES, M. J., NICHOLS, N. K., AND SAMUELS, P. G. Steady Open Channel Test Problems with Analytic Solutions. Numerical analysis report 3/95, University of Reading, Department of Mathematics, 1995.

[17] MESELHE, E. A., AND HOLLY, F. M. Invalidity of Preissmann Scheme for Transcritical Flow. *Journal of Hydraulic Engineering 123*, 7 (1997).

[18] MESELHE, E. A., SOTIROPOULOS, F., AND HOLLY, F. M. Numerical simulation of transcritical flow in open channels. *Journal of Hydraulic Engineering 123*, 9 (1997).

[19] MITCHELL, S. L. *Coupling Transport and Chemistry: Numerics, Analysis and Applications*. PhD thesis, Department of Mathematical Sciences, University of Bath, 2003.

[20] MORTON, K. W. A preliminary analysis of the Box Scheme for open channel flow. Ucina project - report for hydraulics research, Oxford University Computing Laboratory, 1984.

[21] MORTON, K. W. *Numerical Solution of Convection-Diffusion Problems*. Chapmann and Hall, London, 1996.

[22] MORTON, K. W. Hyperbolic Conservation Laws (Topics in Differential Equations). Lecture Notes, 2003.

[23] MORTON, K. W., AND BURGESS, N. A. The stability of boundary conditions for an angled-derivative difference scheme. *Advances in Computational Mathematics 6* (1996), 263–279.

[24] MORTON, K. W., AND MAYERS, D. F. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, 1994.

[25] MORTON, K. W., RUDGYARD, M. A., AND SHAW, G. J. Upwind iteration methods for the cell vertex scheme in one dimension. *Journal of Computational Physics 114*, 2 (1994), 209–226.

[26] OCKENDON, H., AND OCKENDON, J. R. *Viscous Flow*. Cambridge University Press, 1995.

[27] PREISSMANN, A. Propogation des intumescenes pas les canaux et rivières. In *1st Congr. de l'Assoc. Francaise de Calcul* (1961), Association Francaise de Calcul, Grenoble, France, pp. 433–442.

[28] RICHTMYER, R. D., AND MORTON, K. W. *Difference Methods for Initial-Value Problems*, 2nd ed. Interscience Publishers, New York, 1967.

[29] SKEELS, C. P. *One dimensional river modelling*. PhD thesis, Department of Numerical Analysis, Faculty of Mathematics, University of Oxford, 1992.

[30] SPENCE, A., AND GRAHAM, I. G. Numerical Methods for Bifurcation Problems. Lecture Notes, 2002.

Index