THE UNIVERSITY OF READING

A C-property Satisfying RKDG Scheme with Application to the Morphodynamic Equations

Paul Jelfs
Department of Mathematics
University of Reading
Reading, UK
RG6 6AX

January 2008

This thesis is submitted for the degree of Doctor of Philosophy

Abstract

The morphodynamical equations can be used to model water flow and bed load sediment transport in rivers and coastal regions. They are a hyperbolic inhomogeneous system of conservation laws consisting of the standard shallow water equations and a bed-updating equation. Any numerical scheme that attempts to model these equations must satisfy the C-property. This ensures that the scheme settles to the correct steady state in the absence of flow. Work by Hudson [34] tested numerical schemes in a finite difference setting demonstrating the need for C-property satisfaction. However a model of a physical problem also usually contains a complicated domain that naturally suggests a finite element scheme should be employed.

The Runge-Kutta Discontinuous Galerkin method is a finite element scheme, studied by Cockburn *et al.* [17, 16, 15, 13, 18], that combines a discontinuous spatial discretisation, finite volume style numerical fluxes, TVD Runge-Kutta time stepping and a TVD slope limiter to obtain a high order solution that retains ideal properties and explicitness. Schwanenberg *et al.* [68] showed, through comparisons with physical models, that the RKDG method is a viable solver for the shallow water equations only when the bed is continuously represented.

This research defines an RKDG method that incorporates a finite difference style source discretisation and also accounts for the dual speed nature of morphodynamics. It is applied to the modelling of the morphodynamical equations through the introduction of a discontinuously represented, moving bed. This new, C-property satisfying, scheme is given for 1D and 2D and results are given to demonstrate that, although the standard RKDG method fails to model the morphodynamic equations well it is successful with the proposed scheme.

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Acknowledgements

The work contained in this thesis was funded by a CASE studentship between EPSRC and HR Wallingford LTD. I would like to thank H.R. Wallingford, especially Dr. A. J. Cooper, for their assistance and guidance provided during the undertaking of this studentship.

I would like to thank Dr. P. K. Sweby for his dedication to supervision, guidance and encouragement. Additional assistance was provided by Professor M. J. Baines of the University of Reading and I would like to say thanks for all the guidance and the answering of, what was sometimes very simple, questions.

I would also like to thank the friends and family that have supported me during this studentship through encouragement and motivation. Without their support I could not have made it to the end.

For Sarah . . .

Contents

List of Figures

List of Tables

Symbols

Symbol	Definition
h	Water Depth
u,v	Water Velocity
$Q = Qx = hu$, $Qy = hv$	Discharge/ Momentum of Water
B	Bed Height
q	Bed Flux Function
$\mathbf{U}$	Conservative Analytical solution $\mathbf{U} = [h, Q, B]^T$
$\mathbf{W}$	Numerical Approximation to $\mathbf{U}$
$\mathbf{F}$	Flux Function, x direction
$\mathbf{G}$	Flux Function, y direction
$\mathbf{R}$	Source Term Function
$\mathbf{H}$	Numerical Flux Function
$\mathbf{S}$	Numerical Source Function
A	Jacobian A $= \frac{\partial \mathbf{F}}{\partial \mathbf{W}}$
B	Jacobian B $= \frac{\partial \mathbf{G}}{\partial \mathbf{W}}$
X	Right Eigenvector matrix - $X^{-1}\Lambda X = A$
Λ	Diagonal Eigenvalue matrix - $X^{-1}\Lambda X = A$
$\mathbf{W}_j$	$\mathbf{W}$ Evaluated at x_j
$\mathbf{W}_{j-1/2+}$	$\mathbf{W}$ Evaluated at $x_{j-1/2}$ from the right side
$\nu_{(1)}$	basis function labelled 1
$\mathbf{W}_{(1)}$	the coefficient of the solution corresponding to $\nu_{(1)}$
$\mathbf{W}^n$	the numerical solution at the n^{th} time step

$\mathbf{W}^{(i)}$	the i^{th} Runge-Kutta approximation
Δx	Spatial grid spacing.
Δt	Temporal grid spacing.
J	Total number of spacial grid cells
N	Total number of temporal grid cells
j	Current grid position in space, $x = j\Delta x$
n	Current grid position in space, $t = n\Delta t$
$\|\Delta\|$	Area of the grid cell
A	morphodynamical constant
D	surface elevation constant

Chapter 1

Introduction

The impact of hydrodynamics and morphodynamics on today's society can be varied and wide reaching. There are numerous situations in which the changing flow of rivers and estuaries can drastically affect the way in which we go about our lives. The impact of human development and natural change can often be seen in subtle ways such as the drifting of sand dunes and the erosion of cliff faces. They can also be seen in more substantial ways such as flooding and dam breaching.

Morphodynamical change can also have a significant financial impact. The need to continually dredge shipping channels and docks or the flooding of residential land can be financially crippling. Additionally, human generated changes, such as the insertion of a bridge across a river or the extension of a docking platform can result in devastating ecological effects such as the loss of natural wetlands. It is also possible that natural changes can cause the foundations of man-made structures to be undermined, reducing their structural integrity. As such there is a huge interest in developing an accurate model of morphodynamics that can predict the effects of change.

In being able to model morphodynamical situations accurately we can determine the effects of change before they happen and prepare or even restrict the effects of morphodynamical change. Predicting the impact of any developments not only introduces a level of cost effectiveness in our actions but also allows us to take responsibility for minimising the costs of change.

Morphodynamics is a difficult and complex system to model. It involves the interaction of fast moving water, the hydrodynamics, and slower moving sediment transport. The process of modelling the hydrodynamics alone is complicated by many physical effects such as the geometry of the modelled situation, wind driven waves, currents, Coriolis forces, heating and salinity.

Modelling morphodynamics is further complicated by the fact that we cannot yet identify a simple set of equations that are effective at describing the flows involved. To model morphodynamics we combine and use a well established, simple, model of hydrodynamics and a morphodynamical model. This coupling describes the interaction between the hydrodynamics that drive the morphodynamics and the morphodynamical changes that affect the hydrodynamics.

Previous work by Hudson [34] examined the use of finite difference methods to approximate the hydrodynamical and morphodynamical equations. Hudson provided a comprehensive explanation of the equations and defined several formulations from these equations. He tested various finite difference methods on these formulations and demonstrated that some formulations add spurious features to solutions produced by the model.

Hudson also considered the C-property in relation to the various finite difference schemes that he used. This C-property identified schemes that settled to the correct steady state in the absence of flow. He demonstrated that the source term discretisation had a major influence on the viability of the scheme used and that an inappropriate choice of source term discretisation could be severely detrimental to the results that were produced. Hudson tested schemes that satisfied the C-property and schemes that do not, showing that only C-property satisfying schemes could produce viable morphodynamical results.

Although Hudson was able to extend his schemes from 1D to 2D, he was limited to using a dimensional splitting approach on a rectilinear grid. This limits the applicability of his schemes to domains that are well represented by a rectilinear grid. Since finite elements naturally suits irregular unstructured meshes it is reasonable to expect that a finite element method would be more appropriate for modelling hydrodynamics in 2D. This has prompted the recent research into finite element modelling of hydrodynamics. Finite element methods, however, are not naturally applicable to hyperbolic equations.

Finite elements were originally designed to model situations that were governed by elliptic, not hyperbolic, equations. After the following flurry of research relating to finite elements the first steps were taken to modelling hyperbolic equations with finite elements. These steps included the early work for SUPG, streamline upwind Petrov Galerkin, [36]. Later Cockburn *et al.* [17, 16, 15, 13, 18] picked up on the discontinuous Galerkin finite element method and successfully incorporated it, with a TVD Runge-Kutta time stepping algorithm, into a method of lines approach to the discretisation of hyperbolic equations, opening the way for an explosion of RKDG related applications.

Schwanenberg *et al.* [68] applied the RKDG method to the shallow water equations and demonstrated that RKDG could rival some of the best finite difference approaches in 1D and could be easily implemented in 2D. This sets the stage for the work contained within this thesis. We pick up the RKDG method of Cockburn and demonstrate that the raw form, as applied by Schwanenberg, will only model hydrodynamics accurately under an assumption on the solution and is not a viable solver for the morphodynamical models used by Hudson. We then seek to produce a modified method that accurately models morphodynamics without any assumptions on the solution.

1.1 Recent Research

In this section we shall give a brief overview of the literature available for the range of subjects relevant to this thesis. This overview will include basic level information, current research and areas that the work in this thesis may be expanded into. This section will use terms that have not yet been defined and it is suggested that the reader reads the relevant sections of this thesis before referring to this section.

1.1.1 The Shallow Water and Morphodynamic Equations

The shallow water equations have long been used as a model for a range of physical situations. They are derived from the Navier-Stokes equations and there are many papers and books that give the derivation required along with the assumptions that are made to derive them. Derivations can be found, for example, in [73], [77] and [57].

The shallow water equations have been modelled using a large number of numerical schemes. For example, [24] uses some finite volume Godunov schemes to model sub- and trans-critical flow, [33] uses a finite volume scheme with an accurate source term discretisation, [40] uses a second order accurate finite volume scheme, [4] and [11] use a finite element scheme and [65] uses the discontinuous Galerkin method.

The morphodynamical equations build upon the shallow water equations by including the transport of sediment. They include the modelling of sediment transport through the processes of suspended and bed load transport. Although there is no formal definition of the precise equation(s) available, there are a multitude of empirical formulas that have been defined and which are used in many applications. Examples of these can be found in [29], [70],[76] and [55].

The discretisation of the morphodynamical equations has been approached from many angles and is now becoming a popular topic of research. [78] uses a finite volume method for sediment transport modelling, [20], uses ENO and WENO schemes, [30] applies a finite element technique and [57] uses a discontinuous Galerkin method.

1.1.2 The Runge-Kutta Discontinuous Galerkin Method

The discontinuous Galerkin method has its roots in Neutron transport and the concept behind the discretisation first appears in [64] however it remained in an undeveloped form until Chavant [10] combined it with a simple time discretisation. This formed the basis of the work of Cockburn and Shu, who developed the method in the series of papers [17], [16], [15], [13] and [18] from a scalar 1D equation to a full multi-dimensional system of equations. Today, these stand as the definitive source for the arbitrarily high order accurate Rung-Kutta discontinuous Galerkin scheme.

The scheme has recently gained popularity and the range of applications is vast. Examples are the Euler Equations [52], the level set equation [53], Maxwell's equations [19] and Korteweg-de Vries equation [49].

Additionally, many new developments and extensions of the discontinuous Galerkin method are now available. Examples are particle-in-cell [35], spectral/hp discontinuous Galerkin [23] and moving mesh discontinuous Galerkin [47].

For a full history of the Runge-Kutta discontinuous Galerkin method and a review of the applications and developments that have occurred see [14].

1.1.3 Application of Method to the Equations

Both, continuous and discontinuous Galerkin methods have been extensively applied to the shallow water equations. Examples include [4] and [5] for continuous Galerkin and [41] and [25] for discontinuous Galerkin, with [21] combining both. For finite element methods [21] approaches the issue by using a continuous Galerkin method for the momentum equation and a discontinuous scheme to improve accuracy for the mass equation, however this still avoids the problem of balancing the source and flux terms.

The topic of morphodynamics is slowly gaining interest to researchers of the discontinuous Galerkin method. The topic introduces some new features to the shallow water model such as stiff source terms and multi-phase flow. Research into the use of the discontinuous Galerkin methods with the morphodynamical equations is becoming a popular subject. Examples of recent work includes [57], [44]. [57] considers only the morphodynamical equation in 1D while [44] uses discontinuous Galerkin for the bed and continuous Galerkin for the water flow and a generalised wave formula replaces the mass equation.

1.1.4 Improving Numerical Schemes

There are many methods for improving a numerical scheme's performance. For example, the choice of numerical flux function can have a major impact on the scheme. [7] has compared several numerical flux functions in a finite volume setting. The use of the Lax-Friedrichs numerical flux in a discontinuous Galerkin setting was studied in [67] whilst [61] has made an extensive study the effect of different numerical flux functions for discontinuous Galerkin.

High resolution schemes employing slope limiters have also been analysed. [32] has looked at the process of slope limiting in a finite volume setting.

For discontinuous Galerkin, [8], [62] and [74] all approach the task of improving the performance of the scheme by modifying the limiter, for example [62] uses a WENO, weighted essentially non-oscillatory, limiter with the discontinuous Galerkin method. [42] demonstrates the effect of shock detection through slope limiting in a discontinuous Galerkin setting.

[22] has attempted to improve the accuracy of the discontinuous Galerkin scheme by combining the method with spectral methods and in [23] this is extended to include *hp*-adaptivity.

There are many approaches to discretising the time derivatives in hyperbolic equations and these are extensively covered in the literature. Examples include [28], [26], [27], [69], [66] and [45]. Although the Runge-Kutta discontinous Galerkin method includes a time discretisation method, it is possible to combine the discontinuous Galerkin part of the scheme with other forms of time discretisation. Examples of these include Lax-Wendroff discontinuous Galerkin [60], Taylor discontinuous Galerkin [58] and implicit time stepping discontinuous Galerkin [63].

Another approach to improving the applicability of schemes is the process of operator splitting, a technique where the equations - and thus the discretisation scheme - are split into its component directions. This reduces a multi-dimensional system to a 1D one, simplifying implementation and improving stability at the expense of accuracy. As finite elements is naturally suited to multi-dimensions this appears to be less useful than other techniques.

As finite elements is naturally suited to elliptic equations, an approach is to split the equations into an elliptic and a hyperbolic part. The finite element method can be used to discretise the elliptic part and a suitable other scheme can be used to discretise the hyperbolic part. This enables the approach to use the best of both method for the parts they are most suited to. This, again, is not inherently useful in the setting of the discontinuous Galerkin scheme as the discontinuous Galerkin scheme is designed to handle the hyperbolic parts in a finite element setting, thus negating the need for another scheme.

1.1.5 Advanced Shallow Water Features

As the shallow water equations drive the morphodynamic equations it is important to accurately model the shallow water equations. There are several approaches that have been made to improving this accuracy.

The C-property has recently become a major factor in defining accurate schemes. The original proposal of the C-property was given in [7]. There have been many papers concerning C-property satisfaction in finite volume schemes, [32] and [33], [56] and [51], but most finite element schemes have circumnavigated the issue by restricting themselves to a continuously represented bed profile.

This C-property satisfaction falls under the topic of flux and source term balancing and this topic has been extensively covered in a finite volume setting, for examples see [56], [51], [33] and [9].

In rivers and estuaries the tidal flows can lead to situations where regions either become wet through flooding or dry through falling surface levels. The modelling of these situations requires the need to model changing domains and is called wetting and drying. A popular approach is to create a domain that extends over the dry areas and switch off the cells which are considered dry. Suitable processes for choosing whether a cell is wet or dry, conserving mass and applying suitable boundary conditions between wet and dry areas are popular areas of study at the moment

[5, 43, 31, 41].

Both [5] and [43] also consider the difference between flooding, sub-critical, and dam-break, super-critical, with [5] taking a more practical viewpoint and [43] taking a more mathematical viewpoint. [41] demonstrates the application of discontinuous Galerkin to wetting and drying in the code ADCIRC.

Also, [54] makes use of *hp*-adaptivity to enhance the discontinuous Galerkin method and [65] demonstrates its use on the shallow water equations with dissipation. Finally [25] has demonstrated the use of the discontinuous Galerkin method for the shallow water equations on a sphere and [1] includes contaminant transport.

1.1.6 Method Applications

Research into the concepts behind morphodynamical modelling are currently being driven by industrial needs for the modelling of real world situations. To achieve this modelling the mathematical methods have been combined into several large scale hydrodynamic modelling codes. These include,

- Telemac 3D - Originally a hydrodynamical code, it is able to model using 2D depth-averaged and full 3D shallow water equations. It is now coupled with an independent morphodynamical flow modeller which independently iterates the bed profile in time and acts as a source for the steady state water solver. Telemac currently uses nodal based schemes based on various forms of finite elements such as SUPG, Streamline Upwind Petrov-Galerkin. These schemes have not yet been demonstrated to satisfy the C-property and, as such, could benefit from research such as is in this thesis.

- ADCIRC - The code ADCIRC (ADvanced CIRCulation) has developed over the last twenty years to accommodate new features. It now includes a basic bed evolution model combined with, both, continuous and discontinuous Galerkin methods. ADCIRC boasts 2D depth integrated and 3D equation solvers, equation linearisation, implicit Crank-Nicolson time stepping and wetting and drying.

- AQUASEA - This is a 2D finite element solver for water flow with a transport solver bolted on.

- RMA10 - This code is written in Fortran 77 and solves 1D, 2D and 3D shallow water equations using continuous Galerkin finite element methods combined with an advection routine. It currently supports salinity modelling, sediment transport, temperature modelling and wetting and drying.

1.2 Thesis Layout

The remainder of this thesis is set out as follows. In Chapter 2 the shallow water equations and the morphodynamical equation will be set out. Different analytical equation formulations will be given as models of hydrodynamics and morphodynamics with the relevant details needed to understand the effects of the differences in formulations.

In Chapter 3 some necessary preliminary properties and methods will be discussed including the C-property. This will allow us to simplify the definition of the method to be used. In Chapter 4 the 1D TVD Runge-Kutta discontinuous Galerkin finite element method will be given in its raw form with the details required to successfully implement the method as a first order, second order and high resolution method.

In Chapter 5 we will prove that the RKDG method in its stated form will only satisfy the C-property when an assumption is made on the solution and test results will be provided as evidence of the achievements and failings of the method. In Chapter 6 we will provide two extensions to the RKDG method that have been designed to account for the failings of the method and show that with these extensions the method becomes highly successful in modelling morphodynamics in 1D.

In Chapter 7 the method will be extended to 2D by giving the standard RKDG scheme and the equivalent extensions needed to successfully model morphodynamics in 2D. Finally we will conclude and indicate further work needed in Chapter 8.

Chapter 2

Formulations in 1D

In this chapter we will outline the equations that shall be the basis of the 1D, one-dimensional, analytical model, which we will approximate using the numerical schemes.

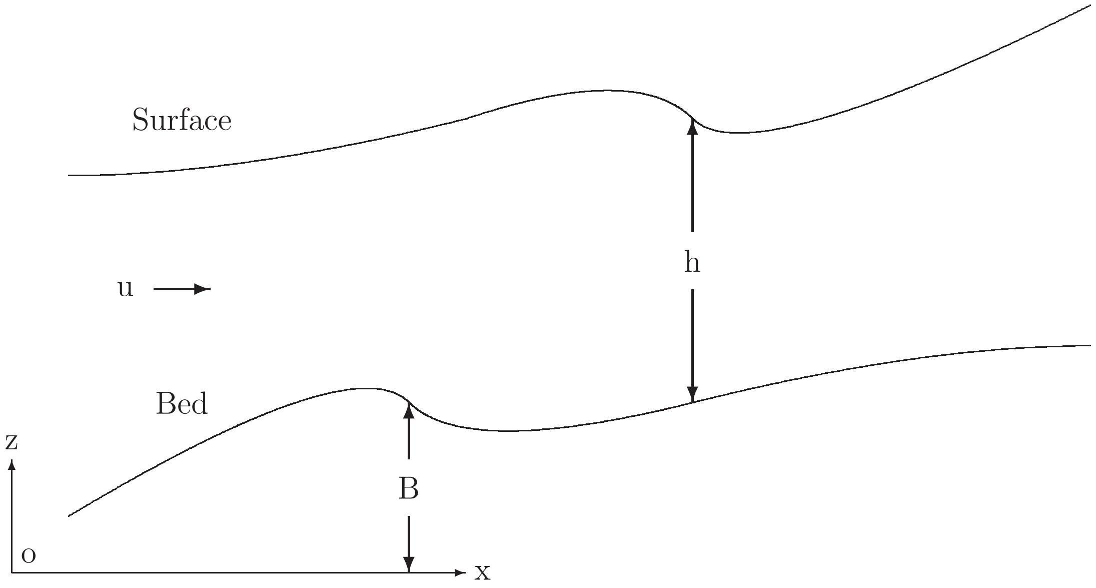


Figure 2.1: The Shallow Water Model

A typical 1D water flow situation is shown in Figure 2.1. In the model, water is flowing along a fixed-width channel over a non-uniform bed. For hydrodynamics the bed remains fixed and for morphodynamics the bed is considered to move. Three equations are used to model the situation and define the hydrodynamics and morphodynamics.

2.1 Hydrodynamics

At every point in the domain we define h, u and B. h is the positive depth of the water, u is a depth averaged horizontal velocity and B is the prescribed elevation of the bed surface relative to some fixed datum point. We make the assumption that the depth is non-zero to keep the water region connected. Therefore, h, u and B are functions of the space variable x.

Additionally, the depth and velocity can change with time and are therefore also a function of the time variable t. For pure hydrodynamics the bed remains fixed in time and so is not a function of t. Prescription of $h = h(x,t)$, $u = u(x,t)$ and $B = B(x)$ will fully describe the flow given in the diagram.

We can also define the discharge $Q = Q(x,t) = h(x,t)u(x,t)$. Prescription of $Q(x,t)$ along with $h(x,t)$ and $B(x)$ will also fully describe the flow since $u(x,t)$ can easily be recovered.

The shallow water equations are a description of the equations governing the water flow shown. They are derived from the Navier-Stokes equations for incompressible flow and make the assumption that the vertical flow speed is negligible compared to the horizontal flow. This assumption means that depth averaging of the velocity gives a much simpler yet still accurate representation of the flow. The shallow water equations are derived by consideration of conservation of mass and momentum in the water flow. Full derivation of these equations has not been repeated here as there are many derivations in the literature such as [73], [57] and [34].

2.1.1 The Mass Conservation Equation

Conservation equations state that the amount of a quantity in a region only changes by the difference in inflow and outflow. It is natural to expect that the amount of water, mass, in a region will only increase if there is a net inflow and decrease if there is a net outflow. We would therefore expect that the change in mass over time will be proportional to the net inflow to, or outflow from, the region.

The mass conservation equation is given by,

$$\frac{\partial}{\partial t}h + \frac{\partial}{\partial x}\Big(hu\Big) = 0,$$

where h and u are given above. This equation can also be written in conservative variable form as

$$\frac{\partial}{\partial t}h + \frac{\partial}{\partial x}Q = 0. \tag{2.1}$$

2.1.2 The Momentum Conservation Equation

In the same manner as the mass conservation, we can expect the momentum in a region to be conserved. The conservation of momentum is affected, however, by forces generated from the bed profile. In the same way as a ball rolling uphill will slow down, we would expect water to slow when the bed is sloping upward. This leads to an equation that is not a true conservation equation when the bed profile is not flat as it includes a forcing, or source, term.

The momentum conservation equation is

$$\frac{\partial}{\partial t}\Big(hu\Big) + \frac{\partial}{\partial x}\Big(\tfrac{1}{2}gh^2 + hu^2\Big) = -gh\frac{\partial B}{\partial x},$$

where h, u and B are given above and g is the gravitational constant of acceleration in the negative vertical, z, direction. This equation can also be written in conservative variable form as

$$\frac{\partial}{\partial t}Q + \frac{\partial}{\partial x}\Big(\tfrac{1}{2}gh^2 + Q^2/h\Big) = -gh\frac{\partial B}{\partial x}. \tag{2.2}$$

The momentum equation can also include other forcing or source terms to account for other features of the model such as friction, Coriolis forces, wind influence or diffusion. For simplicity we shall neglect these terms.

It is also worth noting that, since the model assumes a constant width channel, the momentum and the discharge at any point are equivalent. We will generally refer to the equation as the momentum equation but will refer to the quantity or conserved variable as the discharge.

2.1.3 Non-Conservative Equation

We can combine the mass conservation equation with the momentum conservation equation to give a non-conservative form of this equation,

$$\frac{\partial}{\partial t}u + \frac{\partial}{\partial x}\Big(gh + u^2\Big) = -g\frac{\partial B}{\partial x},$$

but, unfortunately, this equation does not propagate shocks, discontinuities, at the correct speed and so is not considered to be a good approximation. This equation will not be considered in this thesis.

2.1.4 The Hydrodynamic Equations

It is the conservative-variable form of each of the above equations that we shall be considering, i.e. (2.1) and (2.2). These equations together form a hyperbolic system of equations known as the hydrodynamic equations or the shallow water equations, hereafter known as the SWE,

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} hu \\ {}^1\!/\!{}_2 gh^2 + u^2 h \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\frac{\partial B}{\partial x} \end{bmatrix}. \tag{2.3}$$

The derivation of these equations can be found in [34], [73] and [57].

2.1.5 Criticality

We shall see, later in this chapter, that the eigenvalues for the Jacobian of the above system are $u \pm c$ where $c = \sqrt{gh}$ is known as the celerity. These will be of differing signs or the same sign depending on the magnitude of u compared to c. This corresponds to two different states that water can exist in.

If $|u| < c$ then both eigenvalues are of different sign. This is called a sub-critical state and corresponds to calmer water flow. In this situation, waves will travel in both directions, upstream and downstream. An example of this is a river flowing normally.

If $|u| > c$ then both eigenvalues are of the same sign. This is called a super-critical state and corresponds to torrential water flow. In this situation the water is travelling too fast for waves to travel upstream and will only travel downstream. An example of this is a tidal wave.

In the case that $|u| = c$ then we have a critical situation where one of the waves will remain stationary. This is demonstrated by a stationary tidal bore. Generally, in this case we have a different type of flow upstream to that of downstream and this can cause major problems with modelling. In this thesis it will be assumed that all flows are sub-critical as most physical situations that are modelled using these equations are sub-critical. We will also make the assumption that we have a single connected region by enforcing that the depth must always be positive, and non-zero, throughout the domain.

2.2 Morphodynamics

In real life the shape of a river and the profile of the bed will change in time. Both of these processes will alter the flow of the river.

The changing of the shape of the river involves changing the region in which the model is approximated. To accurately approximate this effect we would need to either move our modelled region dynamically over time or somehow account for the entire area that could contain the flow and model this over the entire time span. The first of these methods involves changing the grid which is beyond the scope of this work. The second involves modelling the process of wetting and drying which is, again, beyond the scope of this work.

The changing of the profile of the bed involves the process of scour and deposit. Particles that form the bed mass can be transported by the river, being picked up from one location, known as scour, and dropped at another location, known as deposit. This scour and deposit is achieved by two processes, suspended transport and bed load transport.

Suspended transport is where the bed particles are suspended in the water flow and travel above the bed. This most often happens with smaller particles such as mud particles in fast flowing rivers and gives rise to the discolouration of the water in these rivers. In situations where suspended transport dominates the profile of the bed changes little as there is little scour and deposit occurring.

Bed load transport is where the water is flowing slower or the particle size is too large for the water to suspend the particles. An example of this is sand moving in an estuary. The water tends to be clear and the transported sediment moves slowly compared to the water. Bed load transport is the major factor in scour and deposit. Most of the application areas of morphodynamics involve situations where bed load transport dominates suspended transport. For this reason we will only consider bed load transport.

Hydrodynamic theory assumes a fixed, unmoving bed and the equations governing flow do not account for any change in the bed profile. To take account of bed movement we need to extend the hydrodynamic equations to include an equation describing bed movement. Following the hydrodynamics we can use the principle of conservation of a property to define the equation.

In real life the mass of a bed is preserved. Sand or mud may be moved downstream but none will appear or disappear. We can therefore consider an equation which describes the conservation of bed mass.

This equation is given by,

$$\frac{\partial}{\partial t}B + A\frac{\partial}{\partial x}q(h,u) = 0, \tag{2.4}$$

or,

$$\frac{\partial}{\partial t}B + A\frac{\partial}{\partial x}q(h,Q) = 0,$$

where q is a function of the water properties that describes how the bed changes with respect to water distribution. The equation is known as the sediment transport equation. The parameter A is a constant that accounts for various physical properties such as grain size of the bed material. A realistic value of 10^{-3} has been used here, see [71] and [34]. This function is not given in terms of the bed profile and so is a direct coupling to the hydrodynamics. The function itself is not exactly known and it is very complex, possibly involving the bed too. Various approximations to this function have been proposed, originally by [29], and many are given in [34].

This work will assume a definition of q as,

$$q(h,u) = \xi u^3 = q(h,Q) = \xi Q^3/h^3,$$

as we are only interested in simple bed load transport. The parameter ξ is given as $\xi = 1/(1-\epsilon)$ and ϵ is the porosity of the bed with a realistic value value of 0.4. Derivations of the bed load transport equation and suspended load transport can be found in [70] and [76] and is also given in [57].

2.2.1 The Morphodynamic Equations

The sediment transport equation (2.4), together with the hydrodynamic equations (2.3), form a hyperbolic system of equations known as the morphodynamic equations,

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ uh \\ B \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} hu \\ {}^1\!/_2 gh^2 + u^2 h \\ A\xi u^3 \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\frac{\partial B}{\partial x} \\ 0 \end{bmatrix}. \tag{2.5}$$

It is the overall aim of this thesis to accurately model the morphodynamic equations using a finite element method.

2.3 Formulations

The expression of the hydrodynamic and morphodynamic equations can be achieved in several ways. Different expressions of these equations create different formulations. Hudson [34] proposes several formulations, some of which are repeated here. Each of these formulations are given in the form of,

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F} = \mathbf{R}.$$

The vector $\mathbf{U}$ is known as the vector of conserved variables. In the case of hydrodynamics $\mathbf{U} = [h, Q]^T$. In the case of morphodynamics $\mathbf{U} = [h, Q, B]^T$. The vector $\mathbf{F}$ is known as the flux function and the vector $\mathbf{R}$ is known as the source term.

For each of these formulations we will also give the Jacobian and its eigendecomposition. The Jacobian is a square matrix of the same size as the system and is defined to be $\mathrm{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}$. For this matrix we can decompose it into an eigenvector, X, and eigenvalue matrix, Λ, having the property,

$$\mathrm{A} = \mathrm{X}\Lambda\mathrm{X}^{-1} \qquad \text{or} \qquad \mathrm{A}\mathbf{r}_i = \lambda_i \mathbf{r}_i,$$

where X is the matrix whose columns are the right eigenvectors, $\mathbf{r}_i$, and Λ is the diagonal matrix of corresponding eigenvalues, λ_I. The eigenvalues are also known as the characteristic wave speeds of the system. These essentially tell us the speed at which information is flowing through the domain. We also define here, $|\mathrm{A}| = \mathrm{X}|\Lambda|\mathrm{X}^{-1}$ where $|\Lambda|$ is the diagonal matrix of the modulus of the eigenvalues.

Systems are called strictly hyperbolic if all of the eigenvalues are real and distinct. For all of the following formulations the systems are strictly hyperbolic under the conditions that the flow is subcritical, $|u| < c$, and the depth is positive, $h > 0$.

2.3.1 Formulation SWE-C

The simplest expression of the hydrodynamic equations is as given by (2.3). We therefore define formulation SWE-C to be,

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} hu \\ {}^{1}\!/_{2} g h^2 + u^2 h \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\frac{\partial B}{\partial x} \end{bmatrix}. \tag{2.6}$$

This system has a Jacobian of

$$A = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{bmatrix},$$

which has an eigendecomposition of,

$$\lambda^{(1)} = u - c \qquad\qquad \lambda^{(2)} = u + c$$

$$\mathbf{r}^{(1)} = \begin{bmatrix} 1 \\ u - c \end{bmatrix} \qquad\qquad \mathbf{r}^{(2)} = \begin{bmatrix} 1 \\ u + c \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} u - c & 0 \\ 0 & u + c \end{bmatrix} \quad X = \begin{bmatrix} 1 & 1 \\ u - c & u + c \end{bmatrix}.$$

Later, we shall need the value of |A| under the assumption that $u = 0$, which we determine here for simplicity. We begin with,

$$|\Lambda| = \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix},$$

and,

$$\mathrm{X} = \begin{bmatrix} 1 & 1 \\ -c & c \end{bmatrix}.$$

This means that,

$$\mathrm{X}^{-1} = \frac{1}{2c} \begin{bmatrix} c & -1 \\ c & 1 \end{bmatrix}.$$

From these we can see that,

$$\mathrm{X}|\Lambda| = \begin{bmatrix} c & c \\ -c^2 & c^2 \end{bmatrix}.$$

and,

$$\mathrm{X}|\Lambda|\mathrm{X}^{-1} = \frac{1}{2c} \begin{bmatrix} 2c^2 & 0 \\ 0 & 2c^2 \end{bmatrix}.$$

This means that,

$$|\mathrm{A}| = \mathrm{X}|\Lambda|\mathrm{X}^{-1} = \begin{bmatrix} c & 0 \\ 0 & c \end{bmatrix}. \tag{2.7}$$

2.3.2 Formulation MORPH-C

In the same manner, the simplest expression of the morphodynamic equations is as given by (2.5). We therefore define formulation MORPH-C to be,

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ uh \\ B \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} uh \\ {}^{1\!}/_{2}gh^2 + u^2h \\ A\xi u^3 \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\frac{\partial B}{\partial x} \\ 0 \end{bmatrix}.$$

This system has a Jacobian of

$$A = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 \\ gh - u^2 & 2u & 0 \\ -du & d & 0 \end{bmatrix}$$

where $d = 3A\xi u^2/h$ for convenience. This has an eigendecomposition of

$$\lambda^{(1)} = u - c \qquad \lambda^{(2)} = 0 \qquad \lambda^{(3)} = u + c$$

$$\mathbf{r}^{(1)} = \begin{bmatrix} 1 \\ u - c \\ \frac{-dc}{u-c} \end{bmatrix} \quad \mathbf{r}^{(2)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \mathbf{r}^{(3)} = \begin{bmatrix} 1 \\ u + c \\ \frac{dc}{u+c} \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} u - c & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & u + c \end{bmatrix} \qquad X = \begin{bmatrix} 1 & 0 & 1 \\ u - c & 0 & u + c \\ \frac{-dc}{u-c} & 1 & \frac{dc}{u+c} \end{bmatrix}.$$

This formulation is the same as Formulation A-SF used by Hudson in [34].

Unfortunately, formulation MORPH-C has a zero eigenvalue. This is due to an absence of the bed in the flux function. This gives rise to a Jacobian that has a zero determinant which is demonstrated through the uninvertible eigenvalue matrix, Λ. The two non-zero eigenvalues correspond to those of formulation SWE-C and, indeed, the equations for h and uh are identical. Clearly, none of the eigenvalues, correspondingly the wave speeds, are dependent on the value of A suggesting that any flow modelled with the use of the Jacobian will not be accurate. However, we may find that the inaccuracy is small when the value of A, and thus d, is small so we shall retain and use this formulation for now.

2.3.3 Formulation MORPH-R

To overcome the problems with an uninvertible Jacobian for formulation MORPH-C we can manipulate the equations to avoid this. To perform this manipulation we observe that,

$$-gh\frac{\partial B}{\partial x} = g\frac{\partial h}{\partial x}B - g\frac{\partial hB}{\partial x}.$$

We can replace the source term in this way and move the last term into the flux function to give a flux function that includes the bed profile in its definition. We define formulation MORPH-R to be,

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ uh \\ B \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} uh \\ \tfrac{1}{2}gh(h+2B)+u^2h \\ A\xi u^3 \end{bmatrix} = \begin{bmatrix} 0 \\ gB\frac{\partial h}{\partial x} \\ 0 \end{bmatrix}.$$

We could not use this manipulation in hydrodynamics as it includes a term in B in the flux function. In hydrodynamics the conserved variables do not include B thus the flux function would have included a direct function of x. This would have changed the type of equation that was being solved and would have had severe effects on the methods that could be applied. An example is the derivation of the Roe averages rely on the assumption that the flux function is a direct function of the conserved variables only. In morphodynamics we are entitled to use B in the flux function as it is now a conserved variable.

This system has an invertible Jacobian of

$$A = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 \\ gh+gB-u^2 & 2u & gh \\ -du & d & 0 \end{bmatrix}$$

where $d = 3A\xi u^2/h$ for convenience. Unfortunately, the eigenvalues, and hence the eigenvectors, cannot be explicitly written down. The eigendecomposition is given by,

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \qquad X = \begin{bmatrix} 1 & 1 & 1 \\ \lambda_1 & \lambda_2 & \lambda_3 \\ z(\lambda_1) & z(\lambda_2) & z(\lambda_3) \end{bmatrix}$$

where the eigenvalues are determined from

$$\lambda^3 + (-2u)\lambda^2 + (u^2 - g(h+B) - ghd)\lambda + (guhd) = 0. \tag{2.8}$$

To determine the eigenvalues we use the algorithm shown in Section 2.3.5. The function z is given by,

$$z(\lambda) = \frac{u^2 - g(h+B) - \lambda(2u - \lambda)}{gh}.$$

This formulation is the same as Formulation C used by Hudson in [34].

2.3.4 Formulation SPLIT-C

Since the equations for the hydrodynamics involve larger eigenvalues than the morphodynamical equation there is a natural instinct to disconnect these in some way. There are several benefits to disconnecting the morphodynamical equation including,

- Computationally, the morphodynamic equation can be iterated at its natural speed. This will increase the computational speed of the algorithm.
- By splitting the equations we can eliminate the zero-determinant Jacobian. Since we will be solving the standard hydrodynamic equations the eigendecomposition becomes simple. The only difficulty arising from this method is the calculation of the wave speed associated with the morphodynamics.
- We can, if necessary, modify the method for the morphodynamical equation only; we can benefit from combining different methods.

We, therefore, define formulation SPLIT-C to be,

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ uh \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} uh \\ \frac{1}{2}gh^2 + u^2h \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\frac{\partial B}{\partial x} \end{bmatrix}$$

$$\frac{\partial}{\partial t}B + A\xi\frac{\partial}{\partial x}(u^3) = 0.$$

The hydrodynamic system has a Jacobian and eigendecomposition identical to formulation SWE-C. Since the flux function for the morphodynamic equation does not involve the bed profile we have no direct Jacobian, or wave speed determination. We can define an approximation to the wave speed by using one of the methods given by Hudson in [34]. We choose the same approximation that was used by Hudson in his testing. This gives the wave speed as,

$$\lambda_B = \frac{\partial F}{\partial B} \approx \frac{3A\xi {Q_c}^3}{(D-B)^4},$$

where the discharge is approximately constant at Q_c and the surface is approximately flat and at an elevation of D. Clearly, the accuracy of this wave speed depends on the accuracy of these assumptions. This formulation is the same as Formulation A-CV used by Hudson in [34].

To use a discrete method on these equations we need to determine the interaction times between the hydrodynamic system and the scalar morphodynamic equation. Hudson managed this interaction by performing a single morphodynamic iteration and then followed it with enough hydrodynamic iterations to settle the water to steady state.

Clearly, if the domain, or the effect of the bed change on the hydrodynamics, is large enough then the water will not settle to steady state within the natural time step of the morphodynamics. We, therefore, take the stance that the water will be iterated until it reaches steady state or it is iterated over the time step of the morphodynamics, whichever is the shorter.

The ratio of time steps is determined purely by the wave speeds of the equations.

2.3.5 Eigenvalue Determination

For Formulation MORPH-R we do not have an explicit definition of the eigenvalues. We do however have a cubic polynomial that is satisfied by the eigenvalues. A direct solver for these eigenvalues is given in [34] and [72]. It is approached by writing the polynomial as,

$$\lambda^3 + a_1\lambda^2 + a_2\lambda + a_3 = 0.$$

It should be noted, though, that this algorithm is a costly method for evaluating the eigenvalues and should be used as sparingly as possible. Computationally, eigenvalue determination tends to occur in the heart of a program and is performed a great many times. A single evaluation of the eigenvalues in a numerical scheme can occur at all spatial and temporal grid points and, thus, the total number of evaluations can easily be counted in millions. It is for this reason that we use our knowledge of the system to determine the eigenvalues when we can. If we can explicitly state the eigenvalues then their computational evaluation cost is dramatically decreased, improving the speed of computation. We, therefore, attempt to only use this algorithm when we cannot determine the eigenvalues explicitly.

This algorithm is used to give the roots of a cubic polynomial. A simpler formula exists for the roots of a quadratic equation. For higher orders a more complex algorithm exists and an iterative solver is generally used. The algorithm assumes that the polynomial can be written as shown above where each a is a real number. The eigenvalue polynomial (2.8) is given in the form above and direct comparison gives,

$$\begin{aligned} a_1 &= -2u, \\ a_2 &= u^2 - g(h+B) - ghd, \\ a_3 &= guhd. \end{aligned}$$

We can define,

$$\begin{aligned} S &= {}^1\!/_9({a_1}^2 - 3a_2), \\ R &= {}^1\!/_{54}(9a_1a_2 - 27a_3 - 2{a_1}^3), \\ \theta &= \arccos(R/\sqrt{S^3}). \end{aligned}$$

The eigenvalues are then given by,

$$\begin{aligned} \lambda_1 &= 2\sqrt{S}\cos({}^1\!/_3(\theta + 2\pi)) - {}^1\!/_3a_1, \\ \lambda_2 &= 2\sqrt{S}\cos({}^1\!/_3(\theta + 4\pi)) - {}^1\!/_3a_1, \\ \lambda_3 &= 2\sqrt{S}\cos({}^1\!/_3(\theta + 6\pi)) - {}^1\!/_3a_1. \end{aligned}$$

There is a requirement that $S \geq 0$ since the algorithm uses a square-root operation. In other words we need,

$$S = {}^1\!/_9(4Q^2/h^2 - 3(Q^2/h^2 - g(h+B) - ghd)) \geq 0,$$

or,

$$Q^2/h^2 + 3g(h+B) + 3ghd \geq 0.$$

If this condition is not satisfied then it does not necessarily mean that the eigenvalues cannot be determined, it means that this algorithm cannot be used. An iterative method could find the eigenvalues at the expense of introducing an additional inner iteration process.

Since the sediment transport coefficient, A, is small and all parameters are smooth and continuous we can expect that the eigenvalues given by (2.8) are close to those given by the equation,

$$\lambda^3 + (-2u)\lambda^2 + (u^2 - g(h+B))\lambda + 0 = 0.$$

We would therefore expect to see eigenvalues that are approximately $u \pm c$ and 0. This algorithm will give the eigenvalues in size order when restricted to sub-critical flow only and, although not needed for the eigendecomposition, it does provide a method for diagnosing any problems with the algorithm.

2.4 Summary

In this chapter we have outlined the equations that are the basis of the 1D analytical model, which will be approximated using the numerical schemes. We have presented a formulation for hydrodynamic modelling and three formulations for morphodynamic modelling. In the next chapter we will present some preliminary work in preparation for a detailed breakdown of the numerical schemes.

Chapter 3

Preliminaries

In this chapter we outline some important issues with regard to numerical schemes in general. By outlining these separately we can distinguish what actually forms the numerical scheme themselves without adding generic details. We will consider four properties that can be attained by the scheme and three techniques that can be used with the scheme.

3.1 Properties

For any numerical scheme there are several important properties that it can possess. Any scheme that attains any of these properties will improve our assurance that the scheme will give a good numerical solution to any test case considered. Four properties that we will consider here are conservation, the C-property, Total Variation Diminishing (TVD) and CFL stability limits.

3.1.1 Conservation

The equations that define the SWE model are conservation laws with a source term. They state that the amount of mass and momentum in a region will only change by net inflow and the influence of the source term.

A numerical scheme is said to be conservative if it also has this same property. A conservative scheme will conserve the amount of mass and momentum in the presence of zero difference between inflow and outflow fluxes.

For finite differences any scheme that could be written in the form,

$$W^{n+1} = W^n - \lambda \frac{\Delta t}{\Delta x}(H_{j+1/2} - H_{j-1/2})$$

will be conservative providing,

$$H_{j+1/2} = H(W_{j-p}, \ldots, W_{j+q}) \quad H_{j-1/2} = H(W_{j-p-1}, \ldots, W_{j+q-1}),$$

for some integer constants p and q and a consistent definition of H [45]. An equivalent definition in finite elements is not so readily apparent. To this end we assume that a finite element method is conservative if the finite element discretisation in space, combined with a forward Euler time discretisation is identical to a conservative finite difference scheme.

3.1.2 The C-Property

The C-property is a property first defined by Bermudez & Vasquez [7] which specifically relates to the SWE. It was used to generate finite difference schemes that mimic the analytic case in a simple test scenario. This was used in [34] to generate high-resolution finite difference schemes that also agreed with the analytic case. It is an important requirement for any scheme to satisfy this property for it to give accurate results near, or at, steady state.

Consider a simple test case. The water in the domain of interest is still and the surface is flat. This translates to setting $u = Q = 0$ and $h = D - B$ where D is a constant that specifies the height of the surface. Under this circumstance the analytic equations give a balance between the flux term and source term and we have a zero time derivative. In other words, the solution will not change in time. This balance in the analytic equations gives a simple steady state.

We would also like our numerical scheme to do the same when the numerical solution is approximating the same situation. To express this as a definition:

> Given that $Q{\equiv}0$ and $h{\equiv}D - B$ if the discretisation for the flux term exactly balances the discretisation of the source term, creating a zero time

derivative, then the numerical scheme is said to satisfy the C-property. If this is only true to an order Δx^p then the numerical scheme is said to satisfy the approximate C-property.

Satisfaction of this property requires that the discretisation of the source term exactly balances that of the flux function. If this property is not satisfied then a non-physical numerical solution is inevitable. A common effect of not satisfying the C-property is the generation of a steady state where the depth is constant, not the surface height. If the bed height is not uniform then the water surface will assume the same profile as the bed. This is demonstrated in Figure 3.1.

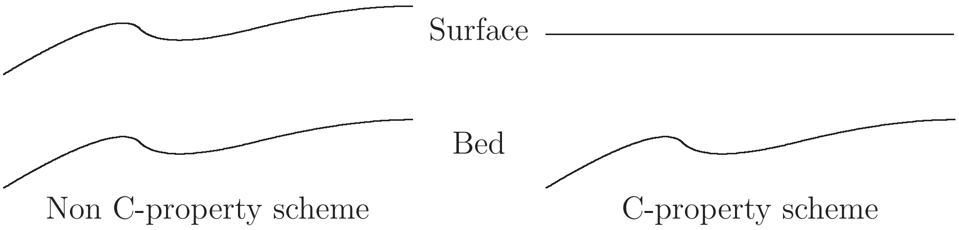


Figure 3.1: Steady States for Non-Uniform Beds

It is worth noting that the definition of the C-property does not stipulate any conditions on the profile of the bed. Most numerical methods are tested with problems that have a flat bed, and thus the source term is cancelled out. This means that the non-compliance with the C-property is not highlighted in these tests. It is important to note that most numerical schemes do not satisfy this property.

3.1.3 Total Variation Diminishing (TVD)

Natural physical dispersion occurs in shallow water flow but the SWE's do not account for it in the numerical model. Although it can be modelled we are not concerned with natural dispersion in this thesis. However, most numerical schemes introduce a numerical dispersion through approximations to the, often, non-linear wave speeds. This dispersion is most visible at shocks and steep slopes [45, pages 119-120], see [77] for further information.

This numerical feature is not natural and can adversely affect the solution. In terms of shallow water, the dispersion causes overshoots and undershoots in the solution near shocks and discontinuities in the form of spurious oscillations and this can cause the depth to become negative. Since the formulation of the equations is based on the assumption that the depth is greater than zero, thus maintaining a connected region, dispersion can create a non-physical approximation. We would therefore like to ensure that the scheme provides a physically meaningful solution.

Maintaining a physically meaningful solution requires, at the very least, minimisation of dispersion in regions where overshoots are likely to occur. An effective way of doing this is to switch to a first order scheme in these regions. This is because the diffusion of first order schemes generally swamps the numerical dispersion. We do not want to excessively use a first order method, however, as this will give a less accurate solution. Since the diffusion is smoothing out the solution, again a non-physical phenomenon, we would like to minimise the use of the first order methods.

High resolution methods employ a first order and a higher order method along with some criteria to determine which, or how much of each, to use in a particular region of the solution. Some methods achieve this through a "tuneable" parameter which allows the method to be tweaked to the particular test problem to achieve the good results. Although this is achievable in simple test cases, where some idea of the exact solution can be used to compare the numerical solutions against, in complex test cases we have little idea of how accurate a tuned solution is.

This suggests that a method which does not require parameter tuning will inspire more confidence in the results it achieves. By creating a method which automatically chooses this tuneable parameter in some optimal way, we can prove that it appropriately balances the diffusion and dispersion and can be confident that the numerical solution is the best obtainable. Modern schemes achieve this automatic balance and we would like our method to do so too.

To prove that a method balances the dispersion and diffusion we need some property that can be evaluated and tested. This property is called total variation diminishing and we can give this a mathematical definition.

The total variation of a scalar, continuous, differentiable function, U, is given by LeVeque, [45], as,

$$TV(U) = \int |\frac{\partial}{\partial x}U| \ dx.$$

We know that this value will never increase for any physically meaningful, non-forced test problem,

$$TV(U(t + \Delta t)) \leq TV(U(t)), \forall \Delta t \geq 0.$$

A numerical equivalent to the analytical definition of TV is given by LeVeque, [45], as,

$$TV(W) = \sum_j |W_{j+1} - W_j|.$$

This definition is consistent as, when $\Delta x \rightarrow 0$ and $W \rightarrow U$, we have,

$$\sum_j |W_{j+1} - W_j| \rightarrow \int |\frac{\partial}{\partial x}U| \ dx.$$

We therefore define a scheme to be total variation diminishing, or TVD, if it satisfies,

$$TV(W^{n+1}) \leq TV(W^n), \forall n.$$

A scheme that satisfies the TVD condition will not suffer from overshoots and thus will give a physically meaningful solution. As an additional note, schemes that are TVD will be, at most, first order accurate at points of extrema in the solution.

There has been a multitude of papers that approach the issue of TVD in finite volumes and finite differences, for example see [9] and [48]. Recently, finite elements has become popular for solving hyperbolic equations and the issue of TVD satisfying finite element methods is still developing.

It is shown in [16] that the RKDG scheme applied to a non-linear homogeneous equation, and with the application of a suitable slope limiter will satisfy the requirements for TVD satisfaction if the solution is taken "in the mean". This means that if we were to consider the mean of the solution in each cell, without the higher orders of reconstruction, then the RKDG scheme is total variation diminishing. We will say that the scheme is total variation diminishing in the mean, TVDM.

Although the TVD property does not naturally extend to systems, we define a system to be TVD if it is component-wise [48].

3.1.4 CFL Limits and Stability

Numerical methods that approximate a hyperbolic model, such as the SWE, have a stability range. This is essentially a requirement on the grid spacing that ensures the method does not "blow up" the numerical solution. This blow up can occur because the numerical scheme does not take into account all of the information that would be present in the analytic case.

For any hyperbolic system information is transmitted through the domain at the characteristic wave speeds. Clearly, if the scheme does not allow the information to travel enough between each time step then we will be missing information in the formulation of the solution. For each point at the new time level we have a spatial domain of dependence at the previous time level. To ensure that the scheme makes use of all of the information required the domain of dependence must, at the very least, contain the characteristics of the problem.

This stability range is dictated by a CFL, Courant-Friedrichs-Lewy, number. If the scheme conforms to being inside this CFL limit then the domain of dependence will contain the characteristics of the problem and it will be stable. If it is considered to be outside the CFL range then we can expect it to tend away from the solution, as it cannot make use of the correct information to define the solution. This commonly appears as blow up toward infinity.

The CFL number is given by the maximum value of,

$$\frac{\max |\lambda_j| \Delta t}{\Delta x},$$

for which the solution will be stable, where λ represents the characteristic wave speeds of the solution over the domain and $\Delta x = \min_j \{\Delta x_j\}$, *i.e.* the smallest grid cell size. Since the spatial grid is predefined for a fixed grid and the wave speeds are defined by the solution, this is normally written as a condition on Δt. We can use this stability condition to derive a condition on the time step that will keep the numerical method stable,

$$\Delta t \leq \frac{\text{CFL}.\Delta x}{\max|\lambda|}.$$

If the time stepping conforms to the above equation then the solution will remain stable and consistent.

3.2 Preliminary Techniques

In addition to these properties we can also define some preliminary numerical techniques. These techniques assist in making any numerical method more accurate. By defining these techniques here we can simplify the definition of the numerical method later. The techniques defined here are adaptive time stepping, non-dimensionalisation and Roe averaging.

3.2.1 Adaptive Time Stepping

With any test problem we will need to approximate it on a numerical grid. This grid will be defined by the parameters Δx and Δt which are the spatial and time grid spacings respectively. Any numerical scheme will have a stability range within which the numerical scheme will be stable and outside which the scheme will not be stable. The stability range is normally given in terms of a CFL number. This number defines the maximum time step that is stable for a given space step,

$$\Delta t = \text{CFL}.\Delta x / \max|\lambda|,$$

where λ is each of the wave speeds for the problem. In practice numerical schemes that use a time step close to this value, although analytically stable, often give spurious instabilities. We wish to make our time steps as large as possible whilst still retaining a stable solution. This is to minimise the amount of computation required. To achieve this we want to get as close to this value as we can, to maximise the time step, without getting too close, so as to introduce instabilities.

In practice the wave speeds of the problem will change as the solution evolves. We could use a uniform time stepping by defining a global Δt and ensure that this remains in the stability range but this will result in more computation than adaptive time stepping. Adaptive time stepping is the process of monitoring the wave speeds of the solution and modifying the time stepping to account for this. In this way we can make large time steps when there is little happening in the solution and the wave speeds are low and make smaller time steps when there are larger waves speeds. We set

$$\Delta t = 0.8\text{CFL}.\Delta x / \max |\lambda|,$$

and use this to set the new time step at each time step. The wave speeds are calculated during computation and the CFL number will be given for each numerical scheme.

We choose the value 0.8 for correspondence with the results that Hudson [34] produced. The value of 0.8 was chosen as this gave numerical results that were devoid of instabilities whilst maximising the time steps. More information about adaptive time stepping can be found in [3].

3.2.2 Non-Dimensionalisation

When modelling shallow water flow, the equations can equally be applied to a bucket of water as they can to an ocean, but the differences in length scales make determining common features difficult. The process of non-dimensionalising the problem removes the element of scale and allows this comparison to be achieved in a more suitable, dimensionless environment. Within this dimensionless environment the effects of the form of the equations are not obscured by the sizes involved and we can distinguish the features that are important from those that are not.

The process of non-dimensionalising involves scaling the various variables by suitably chosen scaling parameters. Although the choice of scaling parameters is not unique, there are some combinations that can provide useful information. Although the process of non-dimensionalisation does not affect the analytical solution it can affect the numerical solution. A side effect of choosing a combination that ensures that Δx is less than 1 is that errors, produced by the inaccuracy of finite precision arithmetic, will be maintained at a lower level and its effect on the solution will be minimised.

For the SWE and morphodynamics we choose to use two length scales, one for space and one for time. We choose the width of the domain as the scaling for space and define L to be the maximum width of the domain. We then redefine our problem by transferring all space variables and parameters into their non-dimensionalised equivalents,

$$\hat{x} = x/L, \qquad \hat{\Delta x} = \Delta x/L, \qquad \hat{h} = h/L, \qquad \hat{B} = B/L,$$

where, in this section only, the “hat” on a symbol represents the non-dimensional form. For the time scaling we are free to choose our scaling in any way that ensures a consistent scheme. Here we choose to use the gravitational constant to define the relationship between the spatial scaling and the time scaling; where T is the timescale, defining $\hat{g} = gT^2/L$ and requiring that $\hat{g} = 1$ implies that

$$T = \sqrt{L/g}.$$

We can then scale the time variables and parameters into their non-dimensionalised equivalents,

$$\hat{t} = t/T, \qquad \hat{\Delta t} = \Delta t/T.$$

We can also combine these scalings to convert all other variables and parameters into their non-dimensionalised equivalents,

$$\hat{Q} = QT/L^2, \qquad \hat{A} = AL/T^2.$$

As an example, Formulation SWE-C is given by (2.6) as

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} hu \\ {}^1\!/\!_2 gh^2 + u^2 h \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\frac{\partial B}{\partial x} \end{bmatrix}.$$

Substituting the above expressions into this gives

$$\frac{\partial}{\partial t}\begin{bmatrix}(\hat{h}/L)\\(\hat{h}/L)(\hat{u}T/L)\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}(\hat{h}/L)(\hat{u}T/L)\\ {}^{1\!}/_{2}(\hat{g}T^2/L)(\hat{h}/L)^2+(\hat{u}T/L)^2(\hat{h}/L)\end{bmatrix}$$

$$=\begin{bmatrix}0\\-(\hat{g}T^2/L)(\hat{h}/L)\frac{\partial(\hat{B}/L)}{\partial x}\end{bmatrix},$$

or

$$\frac{\partial}{\partial t}\begin{bmatrix}\hat{h}/L\\\hat{h}\hat{u}T/L^2\end{bmatrix}+\frac{\partial}{\partial x}\begin{bmatrix}\hat{h}\hat{u}T/L^2\\ {}^{1\!}/_{2}\hat{g}\hat{h}^2T^2/L^3+\hat{u}^2\hat{h}T^2/L^3\end{bmatrix}=\begin{bmatrix}0\\-\hat{g}\hat{h}\frac{\partial\hat{B}}{\partial x}T^2/L^3\end{bmatrix}.$$

However

$$\frac{\partial}{\partial x}=L\frac{\partial}{\partial\hat{x}},\qquad\frac{\partial}{\partial t}=T\frac{\partial}{\partial\hat{t}},$$

giving,

$$\frac{\partial}{\partial\hat{t}}\begin{bmatrix}\hat{h}T/L\\\hat{h}\hat{u}T^2/L^2\end{bmatrix}+\frac{\partial}{\partial\hat{x}}\begin{bmatrix}\hat{h}\hat{u}T/L\\ {}^{1\!}/_{2}\hat{g}\hat{h}^2T^2/L^2+\hat{u}^2\hat{h}T^2/L^2\end{bmatrix}=\begin{bmatrix}0\\-\hat{g}\hat{h}\frac{\partial\hat{B}}{\partial x}T^2/L^2\end{bmatrix}.$$

Multiplying the top equation by L/T and the bottom by L^2/T^2 gives the fully non-dimensionalised form,

$$\frac{\partial}{\partial\hat{t}}\begin{bmatrix}\hat{h}\\\hat{h}\hat{u}\end{bmatrix}+\frac{\partial}{\partial\hat{x}}\begin{bmatrix}\hat{h}\hat{u}\\ {}^{1\!}/_{2}\hat{g}\hat{h}^2+\hat{u}^2\hat{h}\end{bmatrix}=\begin{bmatrix}0\\-\hat{g}\hat{h}\frac{\partial\hat{B}}{\partial x}\end{bmatrix}.$$

It is clear that the non-dimensionalised form is the same as the dimensional form but using different constants, *i.e.* $\hat{g}$ instead of g.

Except for the effect on rounding errors, the process of adaptive time stepping remains unaffected during this non-dimensionalisation. Re-dimensionalisation can be achieved by using the above identities. All computations accompanying this thesis will be performed on the non-dimensionalised equations to minimise the effect of rounding errors but will be displayed in this thesis in their dimensionalised form.

3.2.3 Roe Averages

There will be times when the numerical method will require the evaluation, or value, of the Jacobian matrix at a discontinuity. Since the solution is discontinuous, at the point in question, the value of the Jacobian will be undefined. We, therefore, will need some approximation to the Jacobian and will use the values of the solution either side of the discontinuity to create this approximation. In other words, we require,

$$\bar{A}(\mathbf{W}_{L}, \mathbf{W}_{R}) = \frac{\partial \mathbf{F}}{\partial \mathbf{W}}(\mathbf{W}_{L}, \mathbf{W}_{R}),$$

where $\mathbf{W}_{L}$ and $\mathbf{W}_{R}$ are the values of the solution either side of the discontinuity and $\bar{A}$ is the averaged Jacobian.

Since we do not want to place conditions on either the solution or the flux function we would like the solution to be independent of the ordering of L and R. In other words, if the values of $\mathbf{W}_{L}$ and $\mathbf{W}_{R}$ are swapped then the average Jacobian will remain the same.

The simplest method of creating an approximation to the Jacobian is to average the solution either side of the discontinuity and use this average solution in the Jacobian. The way in which we average is arbitrary, the simplest being the componentwise arithmetic mean,

$$\bar{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{W}}(\bar{\mathbf{W}}) \qquad \bar{\mathbf{W}} = {}^{1\!}/_{2}(\mathbf{W}_{L} + \mathbf{W}_{R}).$$

This average is simple to calculate and will give suitable results.

However, this average does not account for all the information that is available to us. The arithmetic mean will use the values of $\mathbf{W}_{L}$ and $\mathbf{W}_{R}$ but not $\mathbf{F}(\mathbf{W}_{L})$ and $\mathbf{F}(\mathbf{W}_{R})$. If we can use all of the available information in an intelligent manner we would expect to generate a more accurate average Jacobian. Roe averages do this.

Roe averages define the average Jacobian as the evaluation of the Jacobian at an average solution. It differs from the arithmetic mean in the way in which the average solution is defined. The Roe average, given by [45], is required to satisfy the following three conditions

- Conservation: $\bar{A}(\mathbf{W}_{L}, \mathbf{W}_{R}).(\mathbf{W}_{R} - \mathbf{W}_{L}) = \mathbf{F}(\mathbf{W}_{L}) - \mathbf{F}(\mathbf{W}_{L})$.

- Hyperbolicity: $\bar{A}(\mathbf{W}_{L}, \mathbf{W}_{R})$ is diagonalisable with real eigenvalues.

- Consistency: $\bar{A}(\mathbf{W}_{L}, \mathbf{W}_{R}) \rightarrow A(\bar{\mathbf{W}})$ smoothly as $\mathbf{W}_{L}, \mathbf{W}_{R} \rightarrow \bar{\mathbf{W}}$.

The third of these conditions ensures that the average Jacobian is consistent with the equation being solved. It is through the first of these conditions that we can define the average Jacobian.

For all of the formulations given we use an average state given by

$$\bar{h} = {}^1\!/_2(h_L + h_R),$$

$$\bar{B} = {}^1\!/_2(B_L + B_R),$$

$$\bar{Q} = \bar{h}\frac{Q_L\sqrt{h_R} + Q_R\sqrt{h_L}}{h_L\sqrt{h_R} + h_R\sqrt{h_L}},$$

$$\bar{d} = \frac{A\xi({h_R}^2{Q_L}^2 + h_L h_R Q_L Q_R + {h_L}^2{Q_R}^2)}{{h_L}^{5/2}{h_R}^{5/2}}.$$

We then create the average Jacobian by substituting these average values into the Jacobian. The definition of this average Jacobian satisfies the Roe conditions as given above. It is clear that the average Jacobian is hyperbolic as the Jacobians are hyperbolic. Also, since the average Jacobian is formed from averages in this manner, when $\mathbf{W}_L, \mathbf{W}_R \rightarrow \bar{\mathbf{W}}$ then each of the averages also smoothly tends to its corresponding value. To prove that the averages satisfy the conservation property we derive the averages from this requirement.

Derivation of Roe Averages

To show that the Roe averages defined here ensure that the average Jacobian satisfies the properties given we derive them for Formulation MORPH-R. Other formulations are simplifications of this derivation. The Roe averages are required to satisfy

$$\bar{A}(\mathbf{W}_L, \mathbf{W}_R).(\mathbf{W}_R - \mathbf{W}_L) = \mathbf{F}(\mathbf{W}_L) - \mathbf{F}(\mathbf{W}_L).$$

In terms of Formulation MORPH-R this means,

$$\begin{bmatrix} 0 & 1 & 0 \\ g\bar{h} + g\bar{B} - \bar{Q}^2/\bar{h}^2 & 2\bar{Q}/\bar{h} & g\bar{h} \\ -\bar{d}\bar{Q}/\bar{h} & \bar{d} & 0 \end{bmatrix} \begin{bmatrix} h_R - h_L \\ Q_R - Q_L \\ B_R - B_L \end{bmatrix}$$

$$= \begin{bmatrix} Q_R \\ \tfrac{1}{2}gh_R(h_R + 2B_R) + {Q_R}^2/h_R \\ A\xi {Q_R}^3/{h_R}^3 \end{bmatrix} - \begin{bmatrix} Q_L \\ \tfrac{1}{2}gh_L(h_L + 2B_L) + {Q_L}^2/h_L \\ A\xi {Q_L}^3/{h_L}^3 \end{bmatrix}.$$

The top row of the equation is automatically satisfied. The middle row can be split into three separate conditions and the bottom row gives a fourth condition. These conditions are

$$g\bar{h}(h_R - h_L) = \tfrac{1}{2}g{h_R}^2 - \tfrac{1}{2}g{h_L}^2$$

$$g\bar{B}(h_R - h_L) + g\bar{h}(B_R - B_L) = gh_R B_R - gh_L B_L$$

$$-\bar{Q}^2/\bar{h}^2(h_R - h_L) + 2\bar{Q}/\bar{h}(Q_R - Q_L) = {Q_R}^2/h_R - {Q_L}^2/h_L$$

$$-\bar{d}\bar{Q}/\bar{h}(h_R - h_L) + \bar{d}(Q_R - Q_L) = A\xi {Q_R}^3/{h_R}^3 - A\xi {Q_L}^3/{h_L}^3$$

The first of these gives a condition of,

$$\bar{h} = \tfrac{1}{2}(h_L + h_R).$$

Substitution of this into the second gives a condition of,

$$\bar{B} = \tfrac{1}{2}(B_L + B_R).$$

Substituting into the third requires that $\bar{Q}$ takes one of,

$$\bar{Q} = \begin{cases} \bar{h}\frac{\sqrt{h_R}Q_L + \sqrt{h_L}Q_R}{\sqrt{h_R}h_L + \sqrt{h_L}h_R} \\ \bar{h}\frac{\sqrt{h_L}Q_R - \sqrt{h_R}Q_L}{\sqrt{h_R}h_L - \sqrt{h_L}h_R} \end{cases}.$$

Substitution of these into the last condition requires that $\bar{d}$ takes one of,

$$\bar{d} = \begin{cases} A\xi\frac{{h_R}^2{Q_L}^2 + h_L h_R Q_L Q_R + {h_L}^2{Q_R}^2}{{h_L}^{\frac{5}{2}}{h_R}^{\frac{5}{2}}} \\ -A\xi\frac{{h_R}^2{Q_L}^2 + h_L h_R Q_L Q_R + {h_L}^2{Q_R}^2}{{h_L}^{\frac{5}{2}}{h_R}^{\frac{5}{2}}} \end{cases}.$$

Each of these is achieved when the corresponding choice of $\bar{Q}$ is used. It is clear that the second choice of $\bar{Q}$ gives an average, $\bar{d}$, that is not consistent with the original definition of d and, therefore, contradicts the consistency with $\frac{\partial \mathbf{F}}{\partial \mathbf{W}}$. We therefore choose the first definition of $\bar{Q}$ and the corresponding definition of $\bar{d}$. Of additional note is that we can observe that when $h_L = h_R$ the second choice of $\bar{Q}$ is undefined.

Consistency with Hudson

In [34] Roe averages were defined for the same equations. Hudson defined the averages in terms of $u = Q/h$ and not Q. It would be reasonable to expect that the Roe averages that Hudson defined should be consistent with those defined here. The Roe averages given by Hudson are,

$$\begin{aligned}
\bar{h} &= \tfrac{1}{2}(h_L + h_R) \\
\bar{u} &= \tfrac{\sqrt{h_R}u_R + \sqrt{h_L}u_L}{\sqrt{h_L}+\sqrt{h_R}} \\
\bar{B} &= \tfrac{1}{2}(B_L + B_R) \\
\bar{d} &= A\xi \tfrac{\sqrt{h_L}+\sqrt{h_R}}{\sqrt{h_L}h_R+\sqrt{h_R}h_L}({u_L}^2 + u_L u_R + {u_R}^2)
\end{aligned}$$

We need to show that the definitions of $\bar{u}$ are consistent given the equivalence $Q_i = u_i h_i$, $i = L, R$.

$$\begin{aligned}
\frac{\bar{Q}}{\bar{h}} &= \frac{Q_R\sqrt{h_L} + Q_L\sqrt{h_R}}{h_R\sqrt{h_L} + h_L\sqrt{h_R}} \\
&= \frac{\sqrt{h_L}u_R h_R + \sqrt{h_R}u_L h_L}{\sqrt{h_L}h_R + \sqrt{h_R}h_L} \\
&= \frac{(\sqrt{h_L}u_R h_R + \sqrt{h_R}u_L h_L)(\sqrt{h_L} + \sqrt{h_R})}{(\sqrt{h_L}h_R + \sqrt{h_R}h_L)(\sqrt{h_L} + \sqrt{h_R})} \\
&= \frac{(\sqrt{h_L}\sqrt{h_L}u_R h_R + \sqrt{h_L}\sqrt{h_R}u_L h_L + \sqrt{h_R}\sqrt{h_L}u_R h_R + \sqrt{h_R}\sqrt{h_R}u_L h_L)}{(\sqrt{h_L}h_R + \sqrt{h_R}h_L)(\sqrt{h_L} + \sqrt{h_R})} \\
&= \frac{(h_L u_R h_R + \sqrt{h_L}\sqrt{h_R}u_L h_L + \sqrt{h_R}\sqrt{h_L}u_R h_R + h_R u_L h_L)}{(\sqrt{h_L}h_R + \sqrt{h_R}h_L)(\sqrt{h_L} + \sqrt{h_R})} \\
&= \frac{(h_L u_R\sqrt{h_R}\sqrt{h_R} + \sqrt{h_L}\sqrt{h_R}u_L h_L + \sqrt{h_R}\sqrt{h_L}u_R h_R + h_R u_L\sqrt{h_L}\sqrt{h_L})}{(\sqrt{h_L}h_R + \sqrt{h_R}h_L)(\sqrt{h_L} + \sqrt{h_R})} \\
&= \frac{(\sqrt{h_R}u_R + \sqrt{h_L}u_L)(\sqrt{h_L}h_R + \sqrt{h_R}h_L)}{(\sqrt{h_L}h_R + \sqrt{h_R}h_L)(\sqrt{h_L} + \sqrt{h_R})}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{\sqrt{h_R}u_R + \sqrt{h_L}u_L}{\sqrt{h_L} + \sqrt{h_R}} \\
&= \bar{u}.
\end{aligned}
$$

Thus the definitions of $\bar{u}$ are consistent.

We also need to show that the definitions of $\bar{d}$ are consistent under the same equivalence.

$$
\begin{aligned}
\bar{d} &= A\xi \frac{{h_R}^2{Q_L}^2 + h_L h_R Q_L Q_R + {h_L}^2{Q_R}^2}{{h_L}^{5/2}{h_R}^{5/2}} \\
&= A\xi \frac{{h_R}^2{h_L}^2{u_L}^2 + h_L h_R h_L u_L h_R u_R + {h_L}^2{h_R}^2{u_R}^2}{{h_L}^{5/2}{h_R}^{5/2}} \\
&= A\xi \frac{{h_L}^2{h_R}^2({u_L}^2 + u_L u_R + {u_R}^2)}{{h_L}^{5/2}{h_R}^{5/2}} \\
&= A\xi \frac{1}{\sqrt{h_L}\sqrt{h_R}}({u_L}^2 + u_L u_R + {u_R}^2) \\
&= A\xi \frac{\sqrt{h_L} + \sqrt{h_R}}{(\sqrt{h_L} + \sqrt{h_R})(\sqrt{h_L}\sqrt{h_R})}({u_L}^2 + u_L u_R + {u_R}^2) \\
&= A\xi \frac{\sqrt{h_L} + \sqrt{h_R}}{\sqrt{h_L}h_R + \sqrt{h_R}h_L}({u_L}^2 + u_L u_R + {u_R}^2).
\end{aligned}
$$

Thus the definitions of $\bar{d}$ are equivalent. We can, therefore, say that the definitions of the Roe averages are consistent.

3.3 Summary

In this chapter we have given brief details on four properties and three methods that can be applied to any numerical scheme that is approximating the SWE. This now allows us to progress to describing the numerical methods used in this thesis. In the next chapter we will provide an extensive definition of the numerical schemes we have used in 1D.

Chapter 4

The TVD RKDG Finite Element Method

The standard Galerkin finite element method can be applied to the SWE to discretise in space. When combined with a forward Euler time discretisation it can provide a numerical solution that is at least second order accurate in space and first order in time. Unfortunately, the standard Galerkin finite element method has no built in method for compensating the numerical dispersion which occurs due to the high order accuracy. This dispersion causes instabilities with regard to the morphodynamical equations. We are, therefore, forced to seek an improved finite element method to provide a suitable solution.

A method for overcoming the dispersive effects is to add a numerical diffusion, or artificial viscosity, term to the equations. This can alternatively be seen as the diffusion correction in the SUPG, streamline upwind Petrov-Galerkin, method. Although this correction can counter the dispersive effects it introduces too much diffusion and can obliterate the physical phenomena in the numerical solution. There is also the issue of appropriateness as the full, unsimplified SWE include the natural effect of physical diffusion. It can easily be the case that the physical diffusion is minimal yet we need a large, over-dominating, numerical diffusion.

In considering finite element methods that do not introduce numerical diffusion we find that a class of methods that has recently become popular is the TVD RKDG methods. We will consider these as our viable numerical method.

4.1 An Overview of the TVD RKDG Method

The Discontinuous Galerkin, DG, method was first applied to the steady-state neutron transport equations by Read and Hill in 1973 [64]. This was a steady state, elliptic equation. The main advantage of the DG method was that the solution was explicit when a fully upwinded numerical flux was used. The solution could be propagated cell-by-cell from the inflow boundaries to the outflow boundaries.

Following this, there were many papers exploring the application and advancement of the DG method. It has since been successfully applied to a range of applications and combined with many techniques. It is now a powerful contender to most finite volume schemes in two dimensions.

One of the advancements that the DG method underwent was achieved by Cockburn *et al.* in a series of papers [17, 16, 15, 13, 18]. They combined the DG method with a time discretisation to solve hyperbolic equations. Initially a forward-Euler time discretisation was used [17], but this was replaced with a high order Runge-Kutta time discretisation [16] to achieve a uniformly high order method. Extensions were made; to systems [15], to multidimensions [13] and finally to multidimensional systems [18].

DG works on a principle of being a finite element method that does not require continuity between cells. Permitting a discontinuous solution provides a degree of freedom to the Galerkin finite element method. This freedom provides an ability to achieve advances over other Galerkin methods without modifying the equations being solved.

The TVD RKDG method uses a compilation of tools and methods in its definition. Briefly, these are;

- Discontinuous Galerkin Finite Element spatial discretisations.

- TVD Runge-Kutta Finite Difference time discretisations.

- Finite Volume style numerical flux functions.
- A TVD slope limiter in space.

Additional considerations in the definition of the method include boundary conditions and source terms. The full TVD RKDG method is designed to numerically approximate homogeneous hyperbolic equations of the form,

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) = \mathbf{0}. \tag{4.1}$$

In the following sections a complete description of the method is provided. The extension, that is currently used, to accurately approximate an inhomogeneous hyperbolic equation with source term,

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U}) = \mathbf{R}, \tag{4.2}$$

is also given.

4.2 Discontinuous Galerkin

The RKDG method approaches the dual space-time discretisation necessary by initially discretising in space and then following it with a discretisation in time. The numerical solution is a piecewise continuous polynomial representation of the analytical solution. To provide the solution space we partition the space-time domain into $J + 1$ space cells of size Δx and N time cells of size Δt in time. These cells are enumerated, from 0, using j and n respectively. The cell represented by any particular combination of j and n is

$$\{(x, t) : (j - {}^1\!/\!_2)\Delta x \leq x \leq (j + {}^1\!/\!_2)\Delta x, n\Delta t \leq t \leq (n + 1)\Delta t\}.$$

For the spatial discretisation the DG method is used. DG principally is a finite element method that does not require continuity between cells. Allowing this discontinuity provides a greater degree of freedom to the basic Galerkin finite element method. This freedom provides the ability to achieve improvements over other Galerkin finite methods without requiring the equations, that are to be approximated, to be modified.

In the definition of the method we consider a cell by cell construction of the solution. For this reason we will define the method for a single cell. Where the method uses information outside of the current cell we will identify this.

4.2.1 Basis Elements

As with any finite element method, the numerical solution, $\mathbf{W}(x,t)$ in the cell , is represented as a weighted combination of basis functions $\nu(x)$,

$$\mathbf{W}(x,t) = \sum_i \mathbf{W}_{(i)}(t)\nu_{(i)}(x),$$

where a subscript number in brackets identifies a basis function or the corresponding weight. The set of basis functions should be a set of linearly independent spatial functions. It is important to note that the choice of basis functions does not affect the application of method. Indeed, any two sets of basis functions that cover the same space of functions should provide the same results. In the definition of the method, Cockburn *et al.*, use the set of Legendre polynomials as these diagonalise the elemental mass matrix; this will be shown later.

Legendre Polynomials

The Legendre polynomials are a set of functions, defined on the interval [-1,1] which are orthogonal,

$$\int_{-1}^{1} L_{(i)}L_{(k)}\ dx = \begin{cases} 0 & i \neq k \\ \frac{1}{2i+1} & i = k \end{cases},$$

and also have the property,

$$L_{(i)}(-1) = (-1)^i, \qquad L_{(i)}(1) = 1.$$

Any polynomial function of order p can be uniquely represented by a linear combination of the first p Legendre polynomials. The first four Legendre polynomials are

$$L_{(0)}(x) = 1, \qquad L_{(1)}(x) = x, \qquad L_{(2)}(x) = \tfrac{1}{2}(3x^2 - 1), \qquad L_{(3)}(x) = \tfrac{1}{2}(5x^3 - 3x).$$

Legendre Basis Elements

The first two Legendre polynomials, scaled onto the cell j, are

$$\nu_{(0)} = 1, \qquad \nu_{(1)} = \frac{2}{\Delta x}(x - x'),$$

with x' the position of the cell centre and Δx the width of the cell. Therefore, a linear representation of the solution in a single cell can be expressed as a linear combination of these functions. For a first order method we express our numerical solution as,

$$\mathbf{W}(x,t) = \mathbf{W}_{(0)}(t)\nu_{(0)}(x) = \mathbf{W}_{(0)}(t),$$

and for a second order method we express our numerical solution as,

$$\mathbf{W}(x,t) = \mathbf{W}_{(0)}(t)\nu_{(0)}(x) + \mathbf{W}_{(1)}(t)\nu_{(1)}(x) = \mathbf{W}_{(0)}(t) + \mathbf{W}_{(1)}(t)\frac{2}{\Delta x}(x - x'),$$

where $\mathbf{W}_{j,(0)}$ and $\mathbf{W}_{j,(1)}$ are functions of time only and not space. An additional advantage of expressing the solution in this form is that the basis function value and the numerical solution at the cell boundaries can be easily expressed. For a first order method the boundary values of the cell are,

$$\nu_{\pm 1/2,(0)} = 1,$$

$$\mathbf{W}_{j-1/2} = \mathbf{W}_{j+1/2} = \mathbf{W}_{(0)},$$

and for a second order method,

$$\nu_{\pm 1/2,(0)} = 1, \qquad \nu_{\pm 1/2,(1)} = \pm 1,$$

$$\mathbf{W}_{j-1/2} = \mathbf{W}_{(0)} - \mathbf{W}_{(1)}, \qquad \mathbf{W}_{j+1/2} = \mathbf{W}_{(0)} + \mathbf{W}_{(1)}.$$

Other Basis Elements

Any other choice of basis functions can be used with the DG method. By using different set of basis functions we introduce the need to invert a mass matrix. It will be seen in (4.4) that the Legendre polynomials produce a diagonal elemental mass matrix giving a diagonal full matrix. Any other choice of basis functions would produce a full elemental matrix yet, due to the nature of the DG method, would still give a block-diagonal full matrix. It is worth noting, however, that regardless of the choice of basis function this matrix is constant in time and can be pre-calculated and pre-inverted.

4.2.2 Application of the DG Method

The DG method can be applied to the general form of a system of conservation laws with source term, given by (4.2). We initially consider the method as given by Cockburn *et al.* [15] and its application to a system of homogeneous conservation laws as given by (4.1).

As with any finite element method the equation is first multiplied by a scalar spatial test function $\nu(x)$ and then integrated over a single solution cell,

$$\int_{I_j} \nu \frac{\partial}{\partial t}\mathbf{U}\ dx + \int_{I_j} \nu \frac{\partial}{\partial x}\mathbf{F}(\mathbf{U})\ dx = \mathbf{0},$$

where I_j represents the spatial range, or cell in question, $[x_{j-1/2}, x_{j+1/2}]$. The second term, known as the flux term, is then integrated by parts to transfer the derivative on to the test function,

$$\int_{I_j} \nu \frac{\partial}{\partial t}\mathbf{U}\ dx + [\nu\mathbf{F}(\mathbf{U})]_{I_j} - \int_{I_j} \frac{\partial \nu}{\partial x}\mathbf{F}(\mathbf{U})\ dx = \mathbf{0}.$$

We then replace the analytical solution $\mathbf{U}$ with our numerical approximation $\mathbf{W}$ and the test function, ν, with a basis function, $\nu_{(i)}$, to give

$$\int_{I_j} \nu_{(i)} \frac{\partial}{\partial t}\mathbf{W}\ dx + \left[\nu_{(i)}\mathbf{F}(\mathbf{W})\right]_{I_j} - \int_{I_j} \frac{\partial \nu_{(i)}}{\partial x}\mathbf{F}(\mathbf{W})\ dx = \mathbf{0}. \tag{4.3}$$

Since this is a Galerkin method, the test functions are also the basis functions. In this case both the test functions and the basis functions are the Legendre polynomials.

The Time Derivative

The first term in (4.3) is an integration of the product of the test function with the numerical solution. Since our numerical solution is a linear combination of Legendre polynomials we are actually integrating a product of weighted Legendre functions. For a second order method we have

$$\int_{I_j} \nu_{(i)} \frac{\partial}{\partial t} \mathbf{W} \; dx = \int_{I_j} \nu_{(i)} \frac{\partial}{\partial t} \mathbf{W}_{(0)} \nu_{(0)} + \nu_{(i)} \frac{\partial}{\partial t} \mathbf{W}_{(1)} \nu_{(1)} \; dx.$$

The only terms depending on time are the coefficients $\mathbf{W}_{(i)}$ which are constant in any cell. We can therefore rewrite this as

$$\int_{I_j} \nu_{(i)} \frac{\partial}{\partial t} \mathbf{W} \; dx = \frac{\partial}{\partial t} \mathbf{W}_{(0)} \int_{I_j} \nu_{(i)} \nu_{(0)} \; dx + \frac{\partial}{\partial t} \mathbf{W}_{(1)} \int_{I_j} \nu_{(i)} \nu_{(1)} \; dx.$$

Replacing the test function with the basis functions gives

$$\nu = \begin{bmatrix} \nu_{(0)} \\ \nu_{(1)} \end{bmatrix} : \qquad \int_{I_j} \nu \frac{\partial}{\partial t} \mathbf{W} \; dx = \begin{bmatrix} \Delta x & 0 \\ 0 & \Delta x/3 \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} \mathbf{W}_{(0)} \\ \mathbf{W}_{(1)} \end{bmatrix}. \tag{4.4}$$

The matrix given here is the elemental mass matrix. Any choice of basis functions, other than the Legendre polynomials, would have given a full elemental mass matrix.

The mass matrix given by DG differs from that given by standard Galerkin finite elements. Standard Galerkin produces a matrix that appears almost diagonal; the main diagonal is full and any non-zero off-diagonal entries correspond to neighbouring elements in the problem. Since the connectivity of a mesh is usually sparse the mass matrix is also usually sparse. In 1D this mass matrix is tri-diagonal. The important point is that standard Galerkin creates a matrix of size equal to the number of degrees of freedom in the entire problem and still needs a matrix inversion of some form.

DG creates a block-diagonal global mass matrix and, in 1D, these blocks are of size equal to the order of the method. The coupling between cells is relocated to the right-hand vector. It is much faster to invert a block diagonal matrix than the matrix given by the standard Galerkin method. It is this lack of large matrix inversions that can make the DG method considerably faster than other finite element methods.

The additional calculations needed to perform DG is insignificant compared to the matrix inversion of standard Galerkin. As an additional point, since the mass matrix is constant in time, the inversion can be calculated beforehand and simply applied at each time step.

The Boundary Term

The second term in (4.3) is an evaluation of the flux term and test function on the boundary. The Legendre functions have the property that, when scaled onto a cell,

$$\nu_{-1/2,(i)} = (-1)^i, \qquad \nu_{+1/2,(i)} = 1.$$

This means that this second term can be expressed as

$$\nu = \nu_{(i)} : \qquad [\nu \mathbf{F}]_{I_j} = \mathbf{F}(\mathbf{W}_{j+1/2}) - (-1)^i \mathbf{F}(\mathbf{W}_{j-1/2}).$$

Since the solution is discontinuous at the cell boundaries this means that $\mathbf{W}_{j+1/2}$ and $\mathbf{W}_{j-1/2}$ are undefined. We replace the flux function with a numerical flux function, $\mathbf{H}$, of which there are numerous to choose from,

$$\nu = \nu_{(i)} : \qquad [\nu \mathbf{F}]_{I_j} = \mathbf{H}(\mathbf{W}_{j+1/2-}, \mathbf{W}_{j+1/2+}) - (-1)^i \mathbf{H}(\mathbf{W}_{j-1/2-}, \mathbf{W}_{j-1/2+}),$$

where a minus sign following the spatial index indicates the value approaching from the left and a plus sign indicating the value approaching from the right.

There are a significant number of numerical flux functions that have been defined including the (Local) Lax-Friedrichs function, Godonov function and the HLLC function. A number of these have been used in the RKDG method and [61] provides a good comparison of some of them. It is stated in [12], and demonstrated in [61], that as the order of the method increases the effect of the choice of numerical flux function becomes less significant. Since we are only working to second order we make our choice of numerical flux functions in such a way as to provide good results and is simple to define. Additionally the use of these numerical flux functions maintains the localness of the RKDG schemes, being only dependent on the solution immediately either side. It will be seen later that the choice of numerical flux functions given here actually has a significant impact on the C-property satisfaction as the source term discretisation will need to depend on the flux function chosen.

For a first order method we choose the first order upwinded Roe averaged numerical flux,

$$\mathbf{H}(\mathbf{W}_{L},\mathbf{W}_{R}) = {}^1\!/_2\Big(\mathbf{F}(\mathbf{W}_{L}) + \mathbf{F}(\mathbf{W}_{R}) - |\bar{A}|(\mathbf{W}_{R} - \mathbf{W}_{L})\Big), \tag{4.5}$$

and for a second order method we choose the second order centred Lax-Wendroff Roe averaged numerical flux,

$$\mathbf{H}(\mathbf{W}_{L},\mathbf{W}_{R}) = {}^1\!/_2\Big(\mathbf{F}(\mathbf{W}_{L}) + \mathbf{F}(\mathbf{W}_{R}) - s\bar{A}^2(\mathbf{W}_{R} - \mathbf{W}_{L})\Big), \tag{4.6}$$

with $s = \Delta t/\Delta x$ and A is the Roe averaged Jacobian generated from $\mathbf{W}_{L}$ and $\mathbf{W}_{R}$. Both of these numerical fluxes make use of the Roe averages given in Section 3.2.3.

It is worth noting that the evaluation of this numerical flux can be expensive computationally but we only require the numerical flux to be evaluated once at each boundary. The value can be reused between adjacent cells and for each test function making the number of flux evaluations the same as a finite volume method. Therefore efficient coding can make the evaluation of these terms up to four times more effective for a second order method. For higher orders of accuracy we can actually increase the efficiency of the coding as higher orders do not require any more numerical flux evaluations than lower orders.

$$\nu = \begin{bmatrix} \nu_{(0)} \\ \nu_{(1)} \end{bmatrix} : \qquad [\nu\mathbf{F}]_{I_j} = \begin{bmatrix} \mathbf{H}(\mathbf{W}_{j+1/2-},\mathbf{W}_{j+1/2+}) - \mathbf{H}(\mathbf{W}_{j-1/2-},\mathbf{W}_{j-1/2+}) \\ \mathbf{H}(\mathbf{W}_{j+1/2-},\mathbf{W}_{j+1/2+}) + \mathbf{H}(\mathbf{W}_{j-1/2-},\mathbf{W}_{j-1/2+}) \end{bmatrix}.$$

The Flux Integration

The third term in (4.3) is an integration of the flux term and derivative of a test function. It only appears in methods of second order or higher and not when the first Legendre function, $\nu_{(0)}$, is used as the test function. Unfortunately there is no simple way of expressing the derivative of a Legendre polynomial but the derivative can be easily calculated. This integration term is evaluated by a suitable quadrature which is of at least the order of accuracy of the method. For the DG method given by Cockburn *et al.* Gaussian quadrature is recommended although others have used different methods for approximating the integral, see [53]. For a second order method the Gaussian quadrature gives,

$$\int_{I_j} \frac{\partial \nu_{(i)}}{\partial x} \mathbf{F}\, dx = \frac{\Delta x}{2} \left(\frac{\partial \nu_{(i)}}{\partial x} \mathbf{F}(\mathbf{W})\Big|_{x_{j-1/2\sqrt{3}}} + \frac{\partial \nu_{(i)}}{\partial x} \mathbf{F}(\mathbf{W})\Big|_{x_{j+1/2\sqrt{3}}} \right).$$

When we replace the test functions we get

$$\nu = \begin{bmatrix} \nu_{(0)} \\ \nu_{(1)} \end{bmatrix} : \qquad \int_{I_j} \frac{\partial \nu_{(i)}}{\partial x} \mathbf{F}\, dx = \begin{bmatrix} \mathbf{0} \\ \mathbf{F}(\mathbf{W}_{j-1/2\sqrt{3}}) + \mathbf{F}(\mathbf{W}_{j+1/2\sqrt{3}}) \end{bmatrix}.$$

Recombination

When we consider the overall semi-discrete scheme as given by (4.3) we can replace the terms with the equivalents given in the preceding sections. This gives an overall scheme of

$$\begin{aligned} \nu = \begin{bmatrix} \nu_{(0)} \\ \nu_{(1)} \end{bmatrix} : \qquad & \begin{bmatrix} \Delta x & 0 \\ 0 & \Delta x/3 \end{bmatrix} \frac{\partial}{\partial t} \begin{bmatrix} \mathbf{W}_{(0)} \\ \mathbf{W}_{(1)} \end{bmatrix} \\ & + \begin{bmatrix} \mathbf{H}(\mathbf{W}_{j+½-}, \mathbf{W}_{j+½+}) - \mathbf{H}(\mathbf{W}_{j-½-}, \mathbf{W}_{j-½+}) \\ \mathbf{H}(\mathbf{W}_{j+½-}, \mathbf{W}_{j+½+}) + \mathbf{H}(\mathbf{W}_{j-½-}, \mathbf{W}_{j-½+}) \end{bmatrix} \\ & - \begin{bmatrix} \mathbf{0} \\ \mathbf{F}(\mathbf{W}_{j-1/2\sqrt{3}}) + \mathbf{F}(\mathbf{W}_{j+1/2\sqrt{3}}) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \end{aligned}$$

The two rows of this system are decoupled. We can consider them separately, multiply through and rearrange these equations to get

$$\nu = \nu_{(0)} : \quad \frac{\partial}{\partial t} \mathbf{W}_{(0)} = -\frac{\mathbf{H}(\mathbf{W}_{j+½-}, \mathbf{W}_{j+½+}) - \mathbf{H}(\mathbf{W}_{j-½-}, \mathbf{W}_{j-½+})}{\Delta x} \tag{4.7}$$

$$\begin{aligned} \nu = \nu_{(1)} : \quad \frac{\partial}{\partial t} \mathbf{W}_{(1)} &= -3 \frac{\mathbf{H}(\mathbf{W}_{j+½-}, \mathbf{W}_{j+½+}) + \mathbf{H}(\mathbf{W}_{j-½-}, \mathbf{W}_{j-½+})}{\Delta x} \\ &\quad +3 \frac{\mathbf{F}(\mathbf{W}_{j-1/2\sqrt{3}}) + \mathbf{F}(\mathbf{W}_{j+1/2\sqrt{3}})}{\Delta x}. \end{aligned} \tag{4.8}$$

For a first order scheme we only solve the first of these two equations as we only have one test function $\nu_{(0)}$. For a second order scheme we have to solve for both equations. The CFL limit for the DG discretisation with Euler time stepping is 1 for first order and $\frac{1}{3}$ for second order.

The first of these is the standard form that we would expect a finite volume method to take. This means that the DG scheme can be considered to be a finite volume scheme with higher order corrections. Indeed, any finite volume scheme that fits the form of the first of the equations above will have an equivalent DG scheme.

4.2.3 Domain Of Dependence

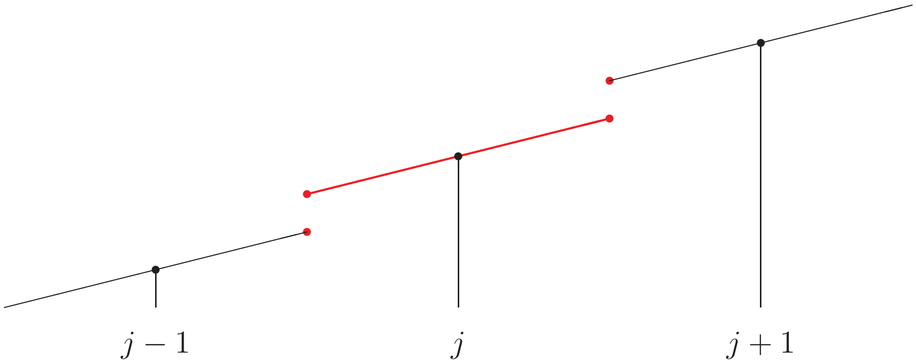


Figure 4.1: Domain of Dependence for DG Discretisation

By considering the semi-discrete discretisation given by (4.7) and (4.8) we can see the spatial dependence of the method. It is clear that the numerical flux functions make use of the discontinuous points on the perimeter of the cell. The numerical quadrature uses the solution inside the cell. Therefore, the solution in a cell is dependent only on the cell itself and the solution immediately outside the cell. This is displayed in Figure 4.1 with the dependence shown in red. We can generalise the DG discretisation by expressing it as,

$$\frac{\partial}{\partial t}\mathbf{W}_{(i)} = \mathbf{L}^{h}_{(i)}(\mathbf{W}, \mathbf{W}_{j-\frac{1}{2}-}, \mathbf{W}_{j+\frac{1}{2}+}). \tag{4.9}$$

This localness of the scheme has several advantages;

- Since the dependence does not propagate into neighbouring cells we have very simple boundary treatment. We need only specify the solution immediately outside the cell to obtain a solution.

- Each cell is contained and the order of approximation in a cell does not depend on the order of approximation in any other cell. We can therefore easily alter the order of approximation, degree of the approximating polynomial, in a cell without having to alter any other cell. This is called p adaptivity.
- The solution in neighbouring cells does not use information about the size or shape of the cell. We can, therefore, split any cell into smaller ones, or merge cells, without affecting any other cell in the grid. This is called h adaptivity.

4.3 TVD Runge-Kutta Time Stepping

The semi-discrete form of the equations given by (4.7) and (4.8) can be written in the form given by (4.9) which we will simplify to

$$\frac{\partial}{\partial t}\mathbf{W} = \mathbf{L}^h(\mathbf{W}), \tag{4.10}$$

where $\mathbf{L}^h(\mathbf{W})$ is an approximation to $-\frac{\partial}{\partial x}\mathbf{F}$. This discretisation, given by the DG method, provides a semi-discrete representation of the solution. It transforms the system of conservation laws into a number of ordinary differential equations that are non-linearly dependent. For a second order method we need to solve two dependent ordinary differential equations in each cell. We therefore discretise in time to obtain a fully discrete numerical approximation. Cockburn *et al.* initially defined the method using the forward Euler time discretisation [17] to get a first order approximation in time but in later papers a TVD form of the Runge-Kutta algorithm was used to obtain a uniformly high order accurate method [16, 15, 13, 18]. These time discretisations are set out below.

4.3.1 Forward Euler

The forward Euler time stepping discretisation is the simplest approximation to (4.10). The derivative is approximated using a first order quadrature

$$\frac{\partial}{\partial t}\mathbf{W} \approx \frac{\mathbf{W}^{n+1} - \mathbf{W}^{n}}{\Delta t}.$$

The right hand side of (4.10) needs to be evaluated at a point in time and the forward Euler discretisation defines this to be at the time given by n. In this way the forward Euler discretisation can be given by,

$$\mathbf{W}^{n+1} = \mathbf{W}^{n} + \Delta t \mathbf{L}^{h}(\mathbf{W}^{n}),$$

where $\Delta t \mathbf{L}^{h}(\mathbf{W}^{n})$ can be considered to be an update to the solution over a single time step. The forward Euler discretisation is a fully explicit method, the solution at any point at time $n+1$ is explicitly given by the solution at time n. Coupled with the DG space discretisation, this gives a fully explicit discretisation for the entire numerical solution.

4.3.2 TVD Runge-Kutta

A TVD form of the Runge-Kutta time stepping discretisation can be applied to the semi-discrete equations given by (4.10). The TVDRK discretisation was proposed as a time stepping method by Shu & Osher [69] and was first coupled with DG by Cockburn *et al.* [16] and its use was retained in the later papers in the series [15, 13, 18]. TVDRK is an adaptation of the standard RK discretisation. TVDRK has the property that it can be proven to retain TVD if the space discretisation provides it.

Whereas the standard RK discretisation is expressed as creating a successively more accurate update between the time steps, TVDRK is expressed as creating successively more accurate approximations to the solution at the next time level. It achieves this by using information gained from the current time step and also the approximations it makes.

The TVDRK algorithm can be summarised by:

- set $\mathbf{W}^{(0)} = \mathbf{W}^n$
- for $i = 1, \dots, k$ compute the intermediate functions

$$\mathbf{W}^{(i)} = \sum_{l=0}^{i-1} \alpha_{il} \mathbf{W}^{(l)} + \beta_{il} \Delta t \mathbf{L}^h(\mathbf{W}^{(l)})$$

- set $\mathbf{W}^{n+1} = \mathbf{W}^{(k)}$

where k is the order of the scheme.

The optimal coefficients α and β are:

$k =$	α_{il}	β_{il}
1	1	1
2	1	1
	$\frac{1}{2}\ \frac{1}{2}$	$0\ \frac{1}{2}$
3	1	1
	$\frac{3}{4}\ \frac{1}{4}$	$0\ \frac{1}{4}$
	$\frac{1}{3}\ 0\ \frac{2}{3}$	$0\ 0\ \frac{2}{3}$

These coefficients maintain the consistency condition that $\sum_l \alpha_{il} = 1$. The CFL limit of the basic scheme is modified by the stability limit of the time discretisation, defined by $\min_{i,l}\{\alpha_{il}/\beta_{il}\}$. It is clear that all of the orders given above have a CFL modifier of 1, thus the CFL limit for the method is given by the CFL limit of the spatial discretisation with Forward Euler time stepping. The combination of the requirement for TVD satisfaction, combined with the maximisation of the CFL limit, gives the optimal choice of coefficients given above.

Unfortunately the coefficients for orders 4 and above require a negative β to obtain the TVD retention property. In these cases an optimal TVD scheme can be found but it requires at least one “backwards” iteration using the adjoint operator of $\mathbf{L}^h(\mathbf{W})$.

It is also worth noting that for a TVD RK scheme we can also guarantee that each RK iteration produces an approximation that is also TVD.

Thus the first order TVDRK algorithm can be expressed as

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \Delta t\mathbf{L}(\mathbf{W}^n)$$

which is equivalent to the forward Euler time stepping algorithm. The second order TVDRK algorithm can be expressed as

$$\begin{aligned}\mathbf{W}^{(1)} &= \mathbf{W}^n + \Delta t\mathbf{L}^h(\mathbf{W}^n)\\ \mathbf{W}^{n+1} &= \tfrac{1}{2}\mathbf{W}^n + \tfrac{1}{2}\mathbf{W}^{(1)} + \tfrac{1}{2}\Delta t\mathbf{L}^h(\mathbf{W}^n).\end{aligned}$$

4.3.3 Other Time Discretisations

Although we have used the TVD Runge-Kutta time discretisation, there are other consistent discretisations that are valid. Examples are the backward, or implicit Euler [63], the standard Runge-Kutta, the implicit Runge-Kutta, Lax-Wendroff [60], the Crank-Nicolson and Leapfrog and Taylor [59]. None of these, however, have been shown to maintain the TVD property given by the discontinuous Galerkin space discretisation.

An alternative to separating the time and space discretisations is to write the equations as a single balance requirement where time, with the spatial dimensions, forms a single set of axes. This space-time discontinuous Galerkin discretisation is implicit but creates a piecewise polynomial representation of the solution in time as well as space. For an examples of this see [50], [75] and [2].

4.4 TVD Slope Limiter

For the TVD RK time stepping to retain the TVD property we need to ensure that the spatial discretisation maintains the TVD property. To ensure this we apply a slope limiter to the solution. Slope limiters modify the information about the slope in a cell to give a representation that has minimal TV. Essentially it is the slope information that makes the method achieve a high order of accuracy. By sensibly modifying the slopes, in regions where dispersion is more likely to pollute the solution, we reduce the accuracy of the method in these regions. The slope limiter given here, a modification of the MUSCL slope limiter, reduces the order of accuracy to one wherever the limiter modifies the solution.

The slope limiter makes use of the MinMod function,

$$m(a,b,c) = \begin{cases} \operatorname{sign}(a)\min\{|a|,|b|,|c|\}, & \text{if } \operatorname{sign}(a) = \operatorname{sign}(b) = \operatorname{sign}(c), \\ 0, & \text{otherwise.} \end{cases}$$

Essentially, if the sign of any of the terms differ then it is zero otherwise it is equal to the smallest in magnitude of the terms.

We will firstly consider a scalar solution and the modify it for systems. To achieve TVD limiting, as given by [17, 16, 15, 13, 18], we first define $\tilde{W}$ and $\tilde{\tilde{W}}$ by,

$$W_{j-1/2+} = W_{(0)} - \tilde{W} \qquad W_{j+1/2-} = W_{(0)} + \tilde{\tilde{W}}.$$

These are essentially the differences between the cell average and the boundary values of the cell. For a first order scheme $\tilde{W} = \tilde{\tilde{W}} = 0$. For a second order scheme $\tilde{W} = \tilde{\tilde{W}} = W_{(1)}$. For higher orders the values of $\tilde{W}$ and $\tilde{\tilde{W}}$ are not so easy to express. By using the cell average and these boundary values we can consider any high order representations of the solution as two linear representations.

We then apply the local projection limiter on these,

$$\tilde{W}^* = m(\tilde{W}, W_{j+1,(0)} - W_{j,(0)}, W_{j,(0)} - W_{j-1,(0)}) \tag{4.11}$$

$$\tilde{\tilde{W}}^* = m(\tilde{\tilde{W}}, W_{j+1,(0)} - W_{j,(0)}, W_{j,(0)} - W_{j-1,(0)}). \tag{4.12}$$

The values passed to each of the m functions are representations of the four distances given as red lines in Figure 4.2. The two outer distances are present in both of the functions whereas the each inner distance only appears in one function. If either $\tilde{W}^* \neq \tilde{W}$ or $\tilde{\tilde{W}}^* \neq \tilde{\tilde{W}}$ then we discard the higher orders, reducing to a linear representation, apply the limiter and take this as the new solution in the cell.

It is worth noting that the limiter uses the solution in the current cell and the cell averages of those surrounding it. This means that the slope in any cell does not affect the limiting of the slopes in any other cell. In addition to making the limiting process highly parallelisable this also ensures that the limiting remains very local.

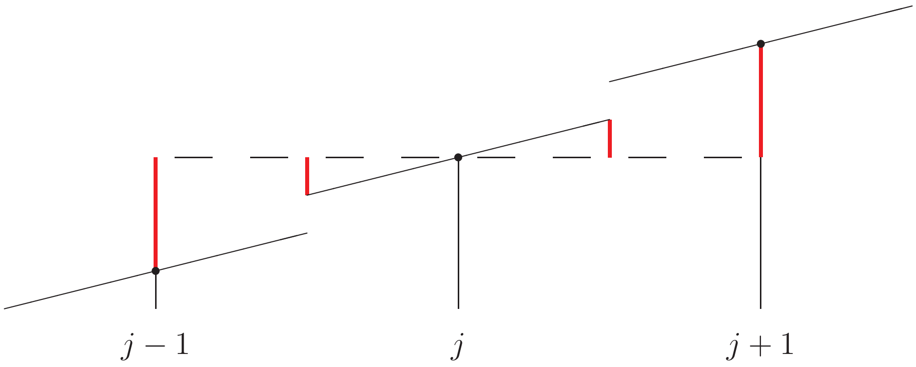


Figure 4.2: Values for the Limiter

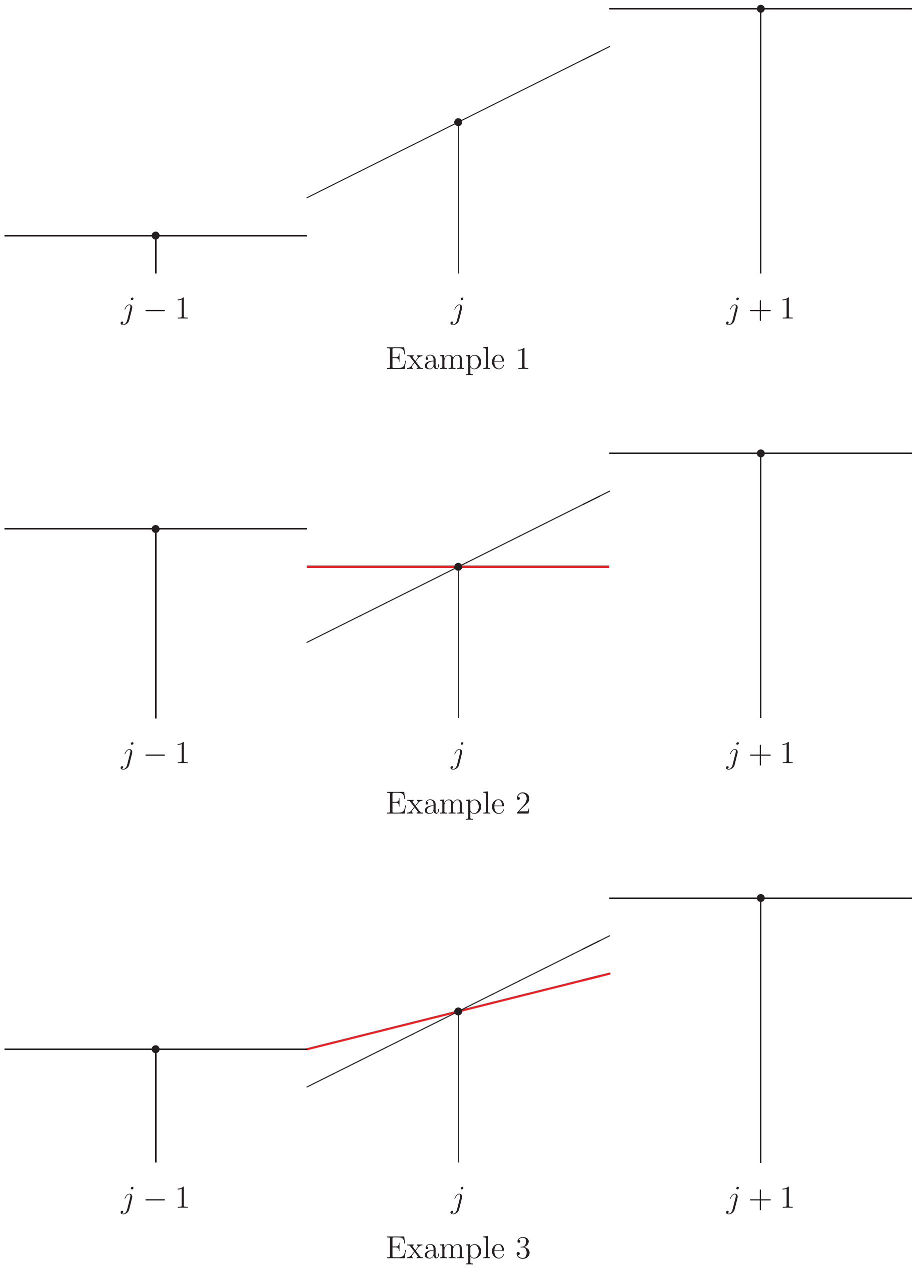
In practice, this localness of the limiter means that shocks are retained within one to two cells, giving a sharp shock profile.

To demonstrate the effect of the limiter some examples of typical values are given in Figure 4.3. The slopes in adjacent cells are shown for guidance and are zero to avoid misinterpretation.

In the case of Example 1 the limiter will not change the solution since the current solution has minimal TV. This is consistent with a smooth slope in smoothly changing regions. Since these regions suffer from little or no dispersion we do not want the limiter to apply here. By not changing the solution we retain the high order of accuracy in these regions.

Application of the limiter to Example 2 and Example 3 would generate a change in solution and this change is given by the red lines; the red line solution is considered to be a better approximation than the black line solution. For Example 2 the solution point is an extremum and it is at extrema that dispersion occurs most. The change to the solution reduces the TV to the minimum whilst retaining conservation.

For Example 3 the solution is equivalent to a sharp change in slope or large second derivative in this cell. The slope given by the difference with the left cell is small compared to the one given by the right cell. This can be created by an incoming shock to which the numerical method would create overshoots in the solution. It is worth noting that the solution in the means is still TVD but this sharp slope change



Example 1

Example 2

Example 3

Figure 4.3: Limiter Examples

will force the solution to lose the TVD property when the shock approaches. The limited slope is the greatest, and nearest to the original, allowable that will retain TVD.

Application of the slope limiter to first and second order methods can be simplified. This is due to the fact that the method is, at most, linear. For a first order scheme a slope limiter is meaningless since the slopes are all zero and the limiter would not have any effect. This is consistent with the fact that a first order scheme is already TVD. We therefore do not limit a first order method. For a second order scheme the solution is piecewise linear and (4.11) and (4.12) are identical. Thus the method simply becomes

$$W^*_{(0)} = W_{(0)}$$

$$W^*_{(1)} = m(W_{(1)}, W_{j+1,(0)} - W_{j,(0)}, W_{j,(0)} - W_{j-1,(0)})$$

where W^* is the limited solution.

For a system of equations the extension of the concept of TVD is not so clear. To remain consistent with the definition of TVD in system form, as see Section 3.1.3, we expand the definition of the scalar limiter as follows. The vector form of the Minmod function is,

$$\mathbf{m}(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \begin{bmatrix} m(a_1, b_1, c_1) \\ m(a_2, b_2, c_2) \\ \vdots \end{bmatrix}.$$

Essentially the MinMod function is applied componentwise. The limiter is then applied to the solution with all limiting being performed componentwise as this has been shown to minimise “wriggles” in the solution [16].

It is worth noting that the limiter is applied spatially only. Since the time stepping is applied to the limited spatial solution the order of accuracy for the time stepping is preserved. In this way, the uniformly high order accurate method only becomes locally first order in a spatial sense without affecting the time discretisation.

The series [17, 16, 15, 13, 18] also proposed a modification of the TVD limiter making it total variation bounded, TVB. In this case, the solution is given some freedom to increase the TV just enough to retain the high order of accuracy. It results in a solution where the TV can grow in time but not blow up. This is achieved by using a modified MinMod function, $\tilde{m}$ given by,

$$\tilde{m}(a,b,c) = \begin{cases} a, & \text{if } |a| \leq M\Delta x^2, \\ m(a,b,c), & \text{otherwise,} \end{cases}$$

where M is some estimate of the second derivative in the region.

Because the TV can grow with the TVB slope limiter the solution is allowed to overshoot, even if only minimally. Since this can cause the unphysical effects in the SWE the TVB slope limiter is not used here. Any use of the limiter will be in the TVD form.

4.4.1 The Shallow Water TVD Limiter

Schwanenberg *et al.* [68], without justification, modified the limiter in the case of the SWE. In this instance the variables that were limited were not strictly the conserved variables. Schwanenberg chose a limiter which limited the surface height, not the depth, in conjunction with the discharge. This is, in some way, more natural as we would like to see a flat surface in most instances. The limiting that Schwanenberg proposes accounts for the bed profile and in most instances produces better results. This is because deep bed profiles allow depth-limited solutions to achieve greater variation than surface-limited solutions. To achieve this limiting, whenever the solution needs to be limited, the surface height is constructed. The limiting is then applied to the surface height and the depth is then recovered.

4.5 Source Terms

The definition of the RKDG method does not account for source terms. In the series of papers by Cockburn *et al.* only homogeneous equations of the form given by (4.1) were considered. Since the SWE include a source term an extension of the method must be made to enable discretisation of inhomogeneous equations of the form given by (4.2). There have been several, more recent, papers that apply the TVD RKDG method to equations with source terms. Most noteworthy, Schwanenberg, [68], used the RKDG method for the shallow water equations. Application of the DG spatial discretisation leaves the term,

$$\int_{I_j} \nu_{(i)} \mathbf{R}(\mathbf{W})\ dx,$$

to be discretised. They use the same quadrature rules for a numerical integration of the source term that is used in numerically integrating the flux function. In other words,

$$\int_{I_j} \nu_{(i)} \mathbf{R}(\mathbf{W})\ dx = \frac{\Delta x}{2}\left(\nu_{(i)}\mathbf{R}(\mathbf{W})\Big|_{x_{j-1/2\sqrt{3}}} + \nu_{(i)}\mathbf{R}(\mathbf{W})\Big|_{x_{j+1/2\sqrt{3}}} \right). \tag{4.13}$$

This gives a final semi-discrete form of

$$\begin{aligned} \nu = \nu_{(0)}: \quad \frac{\partial}{\partial t}\mathbf{W}_{(0)} &= -\frac{\mathbf{H}(\mathbf{W}_{j+\frac{1}{2}-}, \mathbf{W}_{j+\frac{1}{2}+}) - \mathbf{H}(\mathbf{W}_{j-\frac{1}{2}-}, \mathbf{W}_{j-\frac{1}{2}+})}{\Delta x} && (4.14) \\ &\quad +\frac{1}{2}\left(\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}}) + \mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}})\right). \\ \nu = \nu_{(1)}: \quad \frac{\partial}{\partial t}\mathbf{W}_{(1)} &= -3\frac{\mathbf{H}(\mathbf{W}_{j+\frac{1}{2}-}, \mathbf{W}_{j+\frac{1}{2}+}) + \mathbf{H}(\mathbf{W}_{j-\frac{1}{2}-}, \mathbf{W}_{j-\frac{1}{2}+})}{\Delta x} && (4.15) \\ &\quad +3\frac{\mathbf{F}(\mathbf{W}_{j-1/2\sqrt{3}}) + \mathbf{F}(\mathbf{W}_{j+1/2\sqrt{3}})}{\Delta x} \\ &\quad +\frac{\sqrt{3}}{2}\left(-\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}}) + \mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}})\right). \end{aligned}$$

This still can be written in the form given by (4.9) as the source term evaluations only use the solution inside the cell.

4.6 Boundary Conditions

To complete the definition of the TVD RKDG method we need to specify the treatment of boundary conditions. Consideration of the cell immediately adjacent to the boundary suggests that the only information missing at the boundary is the value of the solution immediately outside the domain. If we consider $j = 0$ to identify the leftmost cell in the domain then we would naturally place the numerical boundary at $x_{j-1/2}$. The DG discretisation, given by (4.9), for $j = 0$ is,

$$\frac{\partial}{\partial t}\mathbf{W}_{0,(i)} = \mathbf{L}^h_{(i)}(\mathbf{W}_0, \mathbf{W}_{-1/2-}, \mathbf{W}_{1/2+}).$$

The only value that is undefined is $\mathbf{W}_{-1/2-}$. We use the boundary conditions to set this value. We perform a similar action at the other boundary.

By specifying the solution immediately outside the computational domain we can, through the numerical scheme, determine the entire solution inside the domain. In this way the boundary conditions are weakly imposed through the numerical flux function; the values of the numerical solution at the boundary are not strictly the values given by the boundary conditions. The advantage of this is that it allows simple boundary treatment and outgoing waves pass through the boundary without reflection. Further information about boundary treatment can be found in [37], [38] and [39].

At any boundary the number of boundary conditions needed is equivalent to the number of incoming characteristics. An accurate boundary condition would specify the value of the incoming characteristics, not the value of the solution itself. The advantage of using the numerical flux functions given by (4.5) and (4.6) is that they maintain well-posedness of the problem. The decomposition to characteristics and upwinding in the characteristic plane means that the solution is automatically chosen such that the incoming waves, generated from the specified external state, do not affect the outgoing waves in any way. This, alone, should justify the choice of numerical flux functions given in Section 4.2.2.

In industry the standard method used for determining boundary conditions is multiscaling. The process of multiscaling involves providing an approximation to the flow over a much larger area. For a model of an estuary a larger model may be run that covers the entire ocean or sea that the estuary flows into. This larger model can account for large scale phenomena, provides information on what is occurring in the far-field and can be used to provide approximations to the flow in and around the area of interest. Through the method, specified here, this large scale approximation can be used as consistent boundary information to provide approximate boundary conditions for the model of interest. For this thesis we will assume that a specification of far-field information, otherwise viewed as a large scale approximation to the boundary data, exists, is fully specified and can be provided at any point of interest on our boundary.

4.7 Summary

In this chapter we have given extensive details of the TVD RKDG numerical method in 1D. This numerical method will be the basis of all numerical schemes in this thesis. In the next chapter we will provide proof that the TVD RKDG method as it stands is not a viable numerical method for solving the morphodynamical equations. This will provide motivation for advancing and improving the TVD RKDG method in later chapters.

Chapter 5

Validation

In the previous chapter we gave extensive details of the TVD RKDG method as is described by Cockburn and then applied to the SWE by Schwanenberg. We will now show that this method, in its current form, is not a viable method for approximating the SWE and morphodynamical equations. To show this we will provide a proof that the method does not attain the C-property and show some results for some simple test cases.

5.1 C-Property Proof

We will initially prove that the standard TVD RKDG method, as defined by Cockburn *et al.* and then extended to include the shallow water source term by Schwanenberg, fails the C-property requirement under certain conditions.

To do this we will make the assumptions as given in the definition of the C-property. These are,

$$
\begin{aligned}
Q &\equiv 0 \\
h &\equiv D - B.
\end{aligned}
$$

From these assumptions alone we need to prove that the solution remains at the steady state, i.e. the derivatives with respect to time are zero. We need to show that the spatial discretisation balances the flux discretisation with the source discretisation independently of the time discretisation. To do this we will use the semi-discrete form of the method. We will prove this with the first order method which is also the basis of the higher order methods.

The semi-discrete form of the DG method, with Schwanenberg's source term discretisation is given by (4.14) as,

$$\begin{aligned}\nu = \nu_{(0)}: \quad \frac{\partial}{\partial t}\mathbf{W}_{(0)} &= -\frac{\mathbf{H}(\mathbf{W}_{j+½-}, \mathbf{W}_{j+½+}) - \mathbf{H}(\mathbf{W}_{j-½-}, \mathbf{W}_{j-½+})}{\Delta x} \\ &\quad + \frac{1}{2}\left(\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}}) + \mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}})\right).\end{aligned}$$

To evaluate this we need the numerical flux function for formulation SWE-C under the C-property conditions. The first order Roe-averaged upwind numerical flux gives,

$$\begin{aligned}\mathbf{H}(\mathbf{W}_L, \mathbf{W}_R) &= \frac{1}{2}\begin{bmatrix} 0 \\ ½ g {h_L}^2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 \\ ½ g {h_R}^2 \end{bmatrix} - ½ \begin{bmatrix} \sqrt{g\bar{h}} & 0 \\ 0 & \sqrt{g\bar{h}} \end{bmatrix}\begin{bmatrix} h_R - h_L \\ 0 \end{bmatrix} \\ &= \frac{1}{2}\begin{bmatrix} -\sqrt{½ g(h_L + h_R)}(h_R - h_L) \\ ½ g {h_L}^2 + ½ g {h_R}^2 \end{bmatrix}.\end{aligned}$$

Substituting this into the semi-discrete form gives,

$$\begin{aligned}\frac{\partial}{\partial t}\mathbf{W}_{(0)} &= -\frac{1}{2\Delta x}\begin{bmatrix} -\sqrt{½ g(h_{+-} + h_{++})}(h_{++} - h_{+-}) \\ ½ g {h_{+-}}^2 + ½ g {h_{++}}^2 \end{bmatrix} \\ &\quad + \frac{1}{2\Delta x}\begin{bmatrix} -\sqrt{½ g(h_{--} + h_{-+})}(h_{-+} - h_{--}) \\ ½ g {h_{--}}^2 + ½ g {h_{-+}}^2 \end{bmatrix} \\ &\quad + \frac{1}{2}\left(\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}}) + \mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}})\right),\end{aligned}$$

where subscript ±± represents subscript $j \pm ½\pm$. We also need to evaluate the source terms at the points shown. To simplify calculations these will be written in terms of the end points inside the cell,

$$\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}}) = \frac{g}{6\Delta x}\begin{bmatrix} 0 \\ \left(\sqrt{3}(h_{+-} - h_{-+}) - 3(h_{-+} + h_{+-})\right)(B_{+-} - B_{-+}) \end{bmatrix}$$

$$\mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}}) = \frac{g}{6\Delta x}\begin{bmatrix} 0 \\ \left(-\sqrt{3}(h_{+-} - h_{-+}) - 3(h_{-+} + h_{+-})\right)(B_{+-} - B_{-+}) \end{bmatrix}.$$

We can then replace B in these equations to express the solution in terms of h only using $B = D - h$. This gives source term approximations of,

$$\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}}) = \frac{-g}{6\Delta x}\begin{bmatrix} 0 \\ \left(\sqrt{3}(h_{+-} - h_{-+}) - 3(h_{-+} + h_{+-})\right)(h_{+-} - h_{-+}) \end{bmatrix}$$

$$\mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}}) = \frac{-g}{6\Delta x}\begin{bmatrix} 0 \\ \left(-\sqrt{3}(h_{+-} - h_{-+}) - 3(h_{-+} + h_{+-})\right)(h_{+-} - h_{-+}) \end{bmatrix}.$$

Substituting these into the semi-discrete form gives,

$$\begin{aligned} \tfrac{\partial}{\partial t}\mathbf{W}_{(0)} &= -\tfrac{1}{2\Delta x}\begin{bmatrix} -\sqrt{\tfrac{1}{2}g(h_{+-} + h_{++})}(h_{++} - h_{+-}) \\ \tfrac{1}{2}g{h_{+-}}^2 + \tfrac{1}{2}g{h_{++}}^2 \end{bmatrix} \\ &\quad +\tfrac{1}{2\Delta x}\begin{bmatrix} -\sqrt{\tfrac{1}{2}g(h_{--} + h_{-+})}(h_{-+} - h_{--}) \\ \tfrac{1}{2}g{h_{--}}^2 + \tfrac{1}{2}g{h_{-+}}^2 \end{bmatrix} \\ &\quad -\tfrac{g}{12\Delta x}\begin{bmatrix} 0 \\ \left(\sqrt{3}(h_{+-} - h_{-+}) - 3(h_{-+} + h_{+-})\right)(h_{+-} - h_{-+}) \end{bmatrix} \\ &\quad -\tfrac{g}{12\Delta x}\begin{bmatrix} 0 \\ \left(-\sqrt{3}(h_{+-} - h_{-+}) - 3(h_{-+} + h_{+-})\right)(h_{+-} - h_{-+}) \end{bmatrix}. \end{aligned}$$

It is clear from the top line of this equation that the C-property cannot be satisfied. To achieve a balance we require $h_{++} = h_{+-}$ and $h_{-+} = h_{--}$. This is also true for the bottom line but this is not so apparent. Since $h = D - B$ this equates to the conditions $B_{++} = B_{+-}$ and $B_{-+} = B_{--}$. In other words, the standard DG method, combined with the source term discretisation of Schwanenberg, will satisfy the C-property if and only if the bed is continuously represented.

Higher order methods also do not satisfy the C-property with a discontinuously represented bed. This is because all higher order methods require the solving the same first equation for the means.

Additionally, changing the numerical flux to the second order Roe-averaged centred numerical flux does not solve this problem. Replacement of the same C-property assumptions gives a second order Roe-averaged centred numerical flux of,

$$\mathbf{H}(\mathbf{W}_{\scriptscriptstyle L}, \mathbf{W}_{\scriptscriptstyle R}) = \frac{1}{2}\begin{bmatrix} 0 \\ \tfrac{1}{2}g{h_{\scriptscriptstyle L}}^2 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 0 \\ \tfrac{1}{2}g{h_{\scriptscriptstyle R}}^2 \end{bmatrix} - \tfrac{1}{2}s\begin{bmatrix} g\bar{h} & 0 \\ 0 & g\bar{h} \end{bmatrix}\begin{bmatrix} h_{\scriptscriptstyle R} - h_{\scriptscriptstyle L} \\ 0 \end{bmatrix}$$

$$= \frac{1}{2}\begin{bmatrix} -{}^1\!/\!_2 sg(h_{_L} + h_{_R})(h_{_R} - h_{_L}) \\ {}^1\!/\!_2 g{h_{_L}}^2 + {}^1\!/\!_2 g{h_{_R}}^2 \end{bmatrix},$$

where $s = \Delta t/\Delta x$. This suffers from the same problem of introducing unbalanced terms into the top row of the system of equations. Other numerical fluxes may not introduce the effect of modifying the top line of the equation but will still fail to balance the equations as they will include information from outside the cell. This information will require an assumption to achieve and balance and thus can only satisfy the C-property under this assumption. In general most numerical fluxes will require the same assumption as the first order upwind flux shown above, i.e. that the bed is continuously represented.

In itself the continuously represented bed requirement is not a problem when it comes to modelling pure hydrodynamics. Since the representation of the bed is fixed in the SWE there is no reason that we cannot “choose” the test problem so that the bed representation is continuous. However, having a continuity requirement violates the philosophy of DG.

When we model morphodynamics using DG the bed is represented in a discontinuous manner. Since water flow, with steady inflow, will settle toward a steady state, even if that steady state is constantly changing as in morphodynamics, we can expect that the lack of C-property satisfaction of the standard scheme will result in a poor solution. Due to the proportionally slow movement of the bed, the water flow will see the bed in, pretty much, a steady state. If the water approaches the wrong steady state because the bed is discontinuously represented then we cannot expect the results to be accurate.

5.2 Test Cases

To demonstrate the numerical features of the numerical schemes we will apply them to test cases for which we can qualitatively or quantitatively measure the success of the scheme. The following sections will, firstly, define the test cases and then provide the results for the different schemes.

We shall define all test cases using dimensional variables, specify distance in metres and time in seconds and assume, unless stated otherwise, that $g = 9.81$ for consistency. To perform the calculations themselves the dimensional problem will be non-dimensionalised, calculated and redimensionalised. This will enable us to provide results in dimensional form to aid interpretation. We will additionally use adaptive time stepping. To ensure that we provide output at the specified time, the time step will be suitably adjusted downwards whenever it crosses an output time. Since the scheme is stable for any time step smaller than that given by the CFL condition this practice should be acceptable.

5.2.1 Test Case A

For this test case we will provide the numerical scheme with some initial data that satisfies the assumptions made by the C-property. If a scheme satisfies the C-property then the initial data should create a balance between the discretisations of the flux and source terms and the solution should not change.

$$\begin{aligned} u(x,0) &= 0 \\ h(x,0) &= 10 - B(x,0) \\ B(x,t) &= \begin{cases} 0, & \text{if } \quad 0 \leq x \leq \ 300 \\ \sin^2\left(\pi \frac{(x-300)}{200}\right), & \text{if } 300 \leq x \leq \ 500 \\ 0, & \text{if } 500 \leq x < \ 750 \\ 1, & \text{if } 750 \leq x \leq 1000 \end{cases} \end{aligned}$$

Note the strict inequality on the third term in the specification of B. This means that we can continuously represent the smooth bump on the left but discontinuously represent the step at 750m. We will run the test for for $t = [0, 1000]$ to allow the solution to relax to a steady state if the initial data is not the steady state of the scheme. $1000s$ should allow plenty of time for any generated waves to leave the domain. For the boundary conditions we can specify the flow outside the domain,

$$u(x,t) = 0, \ h(x,t) = 10, \ B(x,t) = 0, \text{ for } x < 0.$$

$$u(x,t) = 0,\ h(x,t) = 9,\ B(x,t) = 1, \text{ for } 1000 < x.$$

This test case can be run on the morphodynamic formulations in addition to the hydrodynamic formulation by setting the parameter $A = 0$.

5.2.2 Test Case B

To demonstrate the ability to model morphodynamics we will define a simple flow over an initially smooth bump in the bed. To ensure that the test problem does not suffer an impulsive start we will lead up to it by allowing the water to reach a steady state before the test is run. This can simply be achieved by running the test problem for a period of time, keeping the bed fixed by setting $A = 0$. We refer to Hudson's test problem B for comparison and run the test for the same time scale. To achieve this we will run the test for $t = [-1000, 540000]$ with the additional requirement that,

$$A = \begin{cases} 0, & \text{if } t{<}0 \\ 0.001, & \text{if } 0{\leq}t \end{cases}.$$

The first period of time, up to $t = 0$, allows the water time to settle down to steady state, the time after models the movement of the bed. We specify the initial data to be,

$$\begin{aligned} u(x,-1000) &= 1 \\ h(x,-1000) &= 10 - B(x,0) \\ B(x,-1000) &= \begin{cases} 0, & \text{if } \quad 0{\leq}x{\leq}\ 300 \\ \sin^2\left(\pi\frac{(x-300)}{200}\right), & \text{if } 300{\leq}x{\leq}\ 500 \\ 0, & \text{if } 500{\leq}x{\leq}1000 \end{cases}. \end{aligned}$$

For the boundary conditions we can specify the flow outside the domain,

$$u(x,t) = 1,\ h(x,t) = 10, \text{ for } x < 0 \text{ and } 1000 < x.$$

Solution to Test Case B

We cannot explicitly determine the analytical solution to this test case at the final time. An approximate solution can be determined up until the point at which characteristics cross [34] but it is not valid after this time. We can, however infer the qualitative character of the solution.

The steady state of the initial solution has a faster water speed at the top of the bed bump than the lower portions. Since the bed flux is a direct function of the water velocity we can, therefore, infer that the flow at the top of the bump is faster than that of the bottom. We would expect the top of the bump to move faster than the bottom.

Since the bottom of the bump on the left side is slower than the top we would expect the region between these points to spread out, making the slope shallower. Also, the bottom of the bump on the right side is slower than the top, so we would expect the right side of the bump should shrink and the slope will steepen. This will create a shock on the leading edge of the bed bump. As the region of greatest slope, on this side, is halfway up the slope, we would expect a shock to form at this point. When the shock forms it will move at a new speed, which should be, due to entropy, faster than the bottom of the bump was moving but slower than the top was moving. Therefore this shock will grow until it consumes the entire right side of the bump.

Until the shock reaches the furthest right extent of the bump, the overall width of the bump should remain the same, however. This is because the front extreme and rear extreme of the bump will both be moving at the same speed. Since the bump starts with a width of $200m$ we would expect the bump to remain at this width until the shock approaches the front of the bump.

As this shock is moving faster than the front extreme of the bump, the bump will increase in width after the shock consumes the front of the bump. As it increases in width it must reduce in height to conserve the volume of bed mass. We should observe a drop in the height of the bump after and we can monitor this progress by evaluating the volume of bed mass in the region. Since, while the bump remains within the region, the inflow and outflow of the bed mass match so the volume should not change. This will also inform us of whether the schemes are actually conservative in practice.

5.3 Results

With all results, the legend identifies the formulation and order of method used. The colour coding and point marking consistently represent the method used. For example, a blue line without point markers represents the first order accurate method applied to formulation MORPH-C.

Order 1 means the first order accurate method using forward Euler or RK1 time stepping and piecewise constant space representation. Order 2U means the second order accurate method using RK2 time stepping and piecewise linear space representation without the application of the limiter. Order 2D means the high resolution method using RK2 time stepping and piecewise linear space representation with the limiter applied to the solution.

If the title of the graph identifies the results as “Means” then this is a representation where the mean value in each cell is used for plotting and any higher order information is discarded. If the graph does not have the “Means” identifier on it then the results are the actual, full representation of the solution that the method produces. We consider the “Means” representation as this is the common form of representation used in finite differences and finite volumes.

5.3.1 Test Case A

The results for test case A for all formulations and the three orders of methods are given in Figure 5.1. It is immediately clear that none of the methods have kept the solution at the correct steady state. The results of the limited methods differ from the unlimited ones by less than 0.01% at $t = 1000$ and so appear the same on the graph.

The results for formulations SWE-C, SPLIT-C and MORPH-C coincide and are given by the dotted lines. This is expected as they are solving the same equations with the same method when the bed is fixed. The results for formulation MORPH-R differ as the equations know a little about the bed profile.

It is clear that all of the first order methods, given by the cyan lines, cannot model the bump on the left. This is due to the fact that the source term is defined in terms of the slope in the cell and for any first order method the slope in the cell is zero, thus the source term must be zero. The second order method and the limited method both include slope information so the source term is not reduced to zero for these methods.

It is clear that none of the methods can model the step on the right side. This is because the bed is discontinuous at this point. Since the source term does not use information about discontinuities between cells it cannot possibly model this.

5.3.2 Test Case B

The results for test case B for all formulations and the three orders of methods are given in Figure 5.2 to Figure 5.4.

For test B we have no results for the second order unlimited methods for formulation MORPH-C and formulation SPLIT-C. This is because it became too oscillatory to remain stable. This can be explained through the fact that the formulations do not have any flux dependence on the discontinuities in B. The limiter appears to be powerful enough to keep the results stable but the oscillations clearly overpower the solution.

For the first order methods for formulation MORPH-C and formulation SPLIT-C, blue lines, we have virtually no movement in the solution. Formulation SPLIT-C has the classic first order diffusion but lacks the transport speed. Formulation MORPH-C has apparently hardly moved the solution and this can be explained by the zero eigenvalue of the formulation.

The second order limited methods for formulation MORPH-C and formulation SPLIT-C, red lines, show a severe change in shape of the bump. They both appear to be “squaring” the profile. This would suggest that it is collecting the discontinuities, that they are failing to model, in one or two locations.

Formulation SPLIT-C, Figure 5.4, with the second order limited method, red dotted line, we can see bed mass transported to the left of the domain. This location

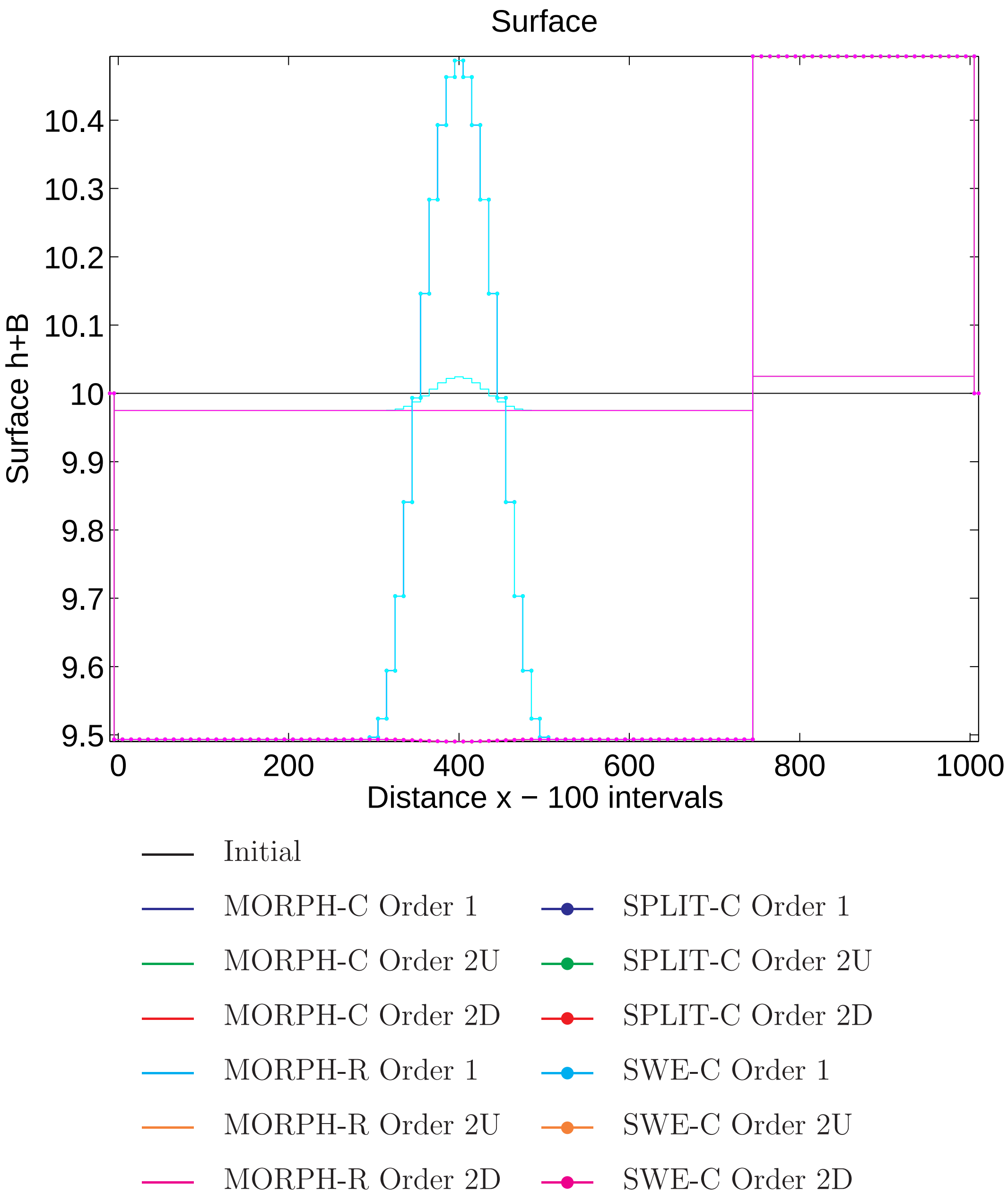


Figure 5.1: Test Case A Results - Standard RKDG

SWE-C, SPLIT-C and MORPH-C Order 1 are coincident and are represented by the cyan dotted line. MORPH-R Order 2U and Order 2D are coincident and are represented by the magenta dotted line. SWE-C, SPLIT-C and MORPH-C, Order 2U and Order 2D are all coincident and are represented by the magenta dotted line.

can be explained by the dual time stepping. Each time the bed moves we have a change in the position that the water would reach if it settled to steady state, this induces a wave on the surface of the water which travels away from the location of the bump in both directions. The time-step size of the bed step allows this water wave to travel approximately $500m$ before the next bed step occurs. Since the bed is free to move, the travelling wave on the water surface induces a wave in the bed profile and this moves directly under the water wave, therefore arriving at the same location. The time step is therefore creating the effect of depositing bed mass at this point. It should be noted that this mass is actually being transported from the bump, with a second, equivalent, amount disappearing out the right boundary.

Formulation MORPH-R, Figure 5.3, appears to be transporting the bump at the correct speed in the manner that we would expect. They all agree on the location of the shock at around $800m$. The first order method, cyan line, is classically diffusive, the second order unlimited method, yellow line, has the classic overshoots leading the shock and the limiter produces a result, magenta line, which has neither of these issues. It is clear that the bump height is slowly reducing as the width increases. This would suggest that formulation MORPH-R is a viable formulation as it seems to have no problems with modelling morphodynamics. One explanation for the apparent success of this formulation, despite the lack of C-property satisfaction is that the water is sufficiently deep to make the water surface profile have minimal effect on the bed movement.

5.4 Summary

It is clear from the results shown above that the TVD RKDG method can only give good numerical results for the SWE under the assumption that the bed is continuously represented. Without this assumption the method cannot satisfy the C-property. In terms of numerics the method cannot model steps in the solution, however small, and first order methods cannot model smoothly represented solutions either.

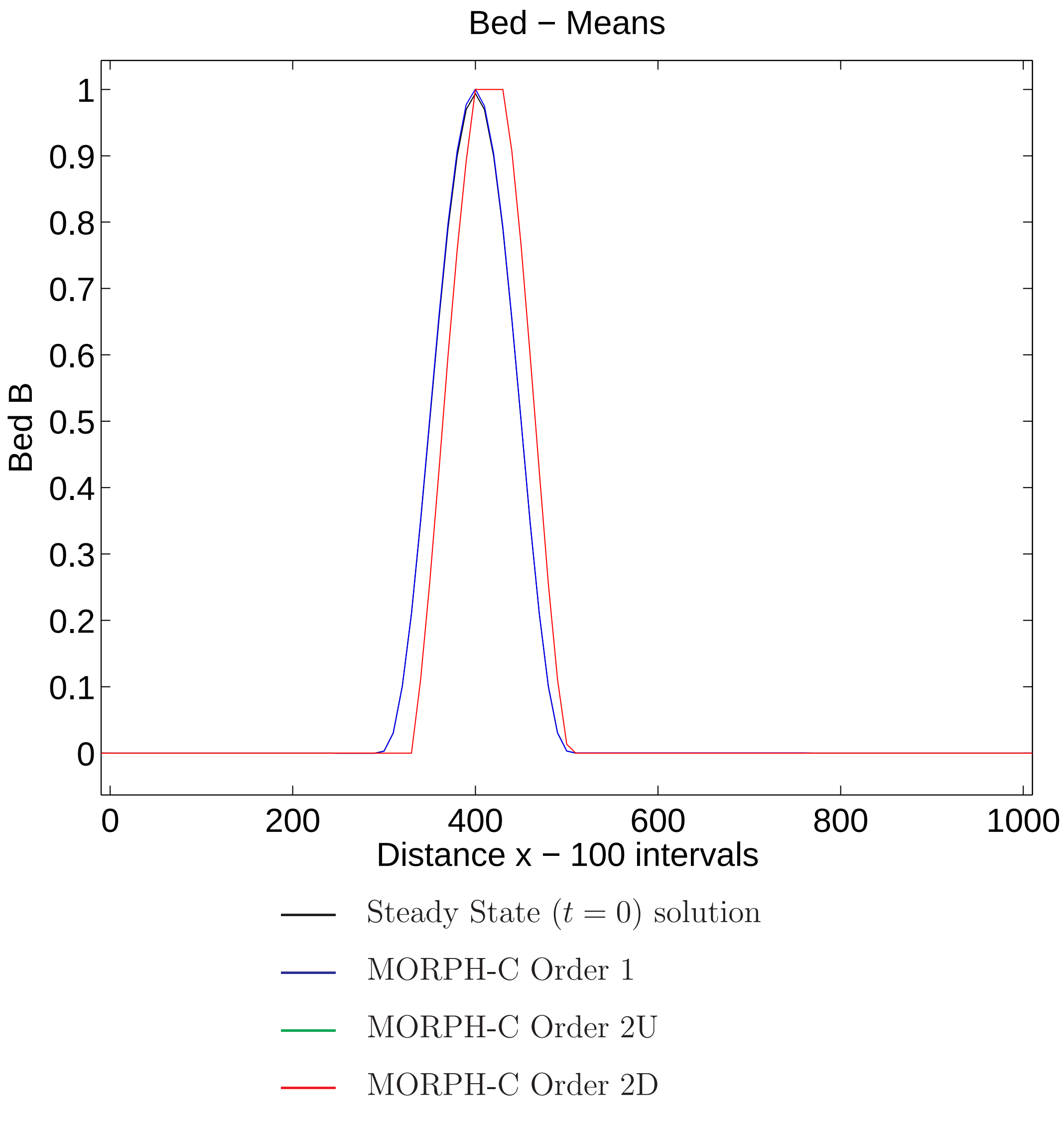


Figure 5.2: Test Case B Results - Standard RKDG with Formulation MORPH-C Note Order 2U is not shown as it failed to complete, causing a negative depth error.

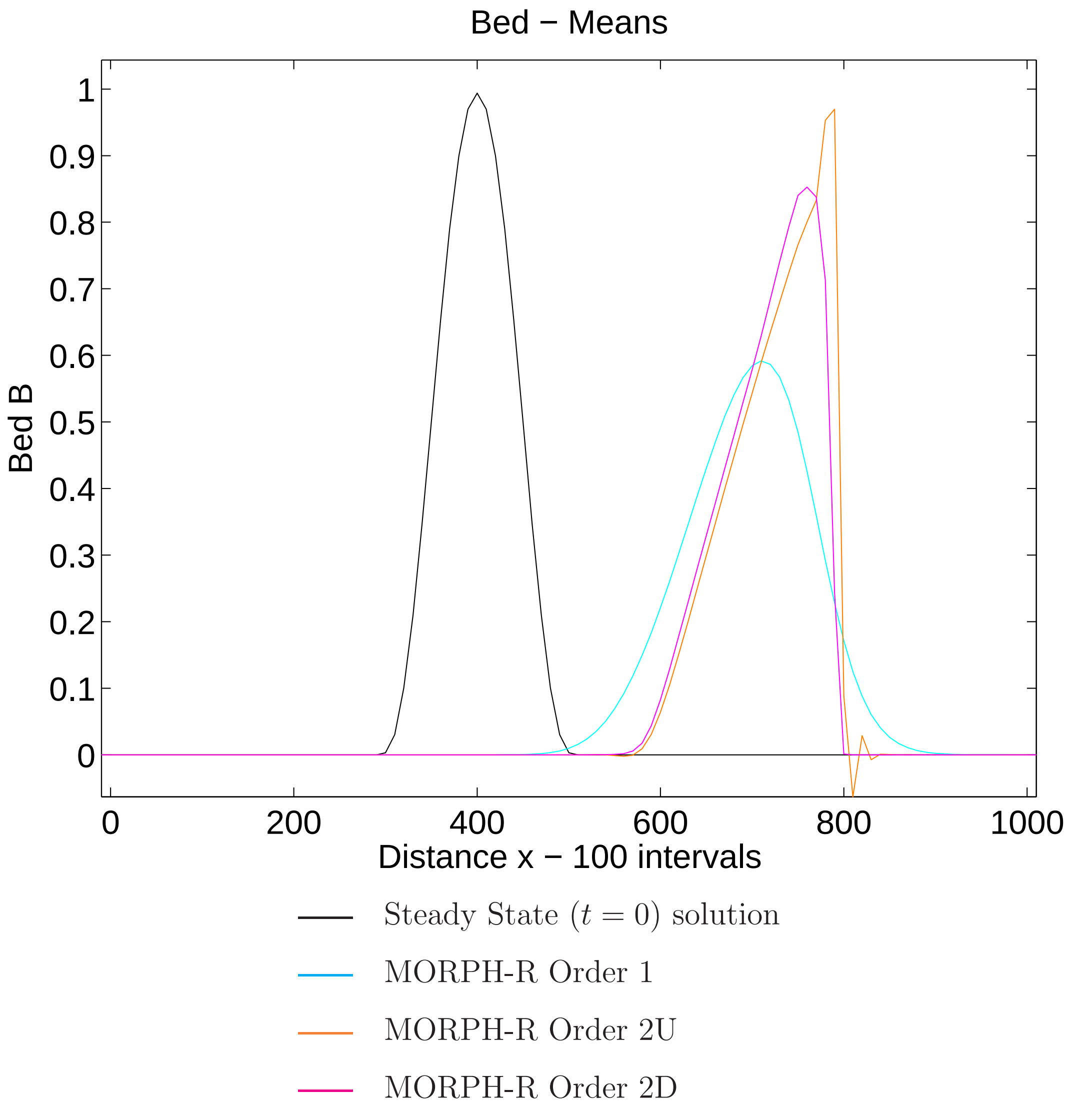


Figure 5.3: Test Case B Results - Standard RKDG with Formulation MORPH-R

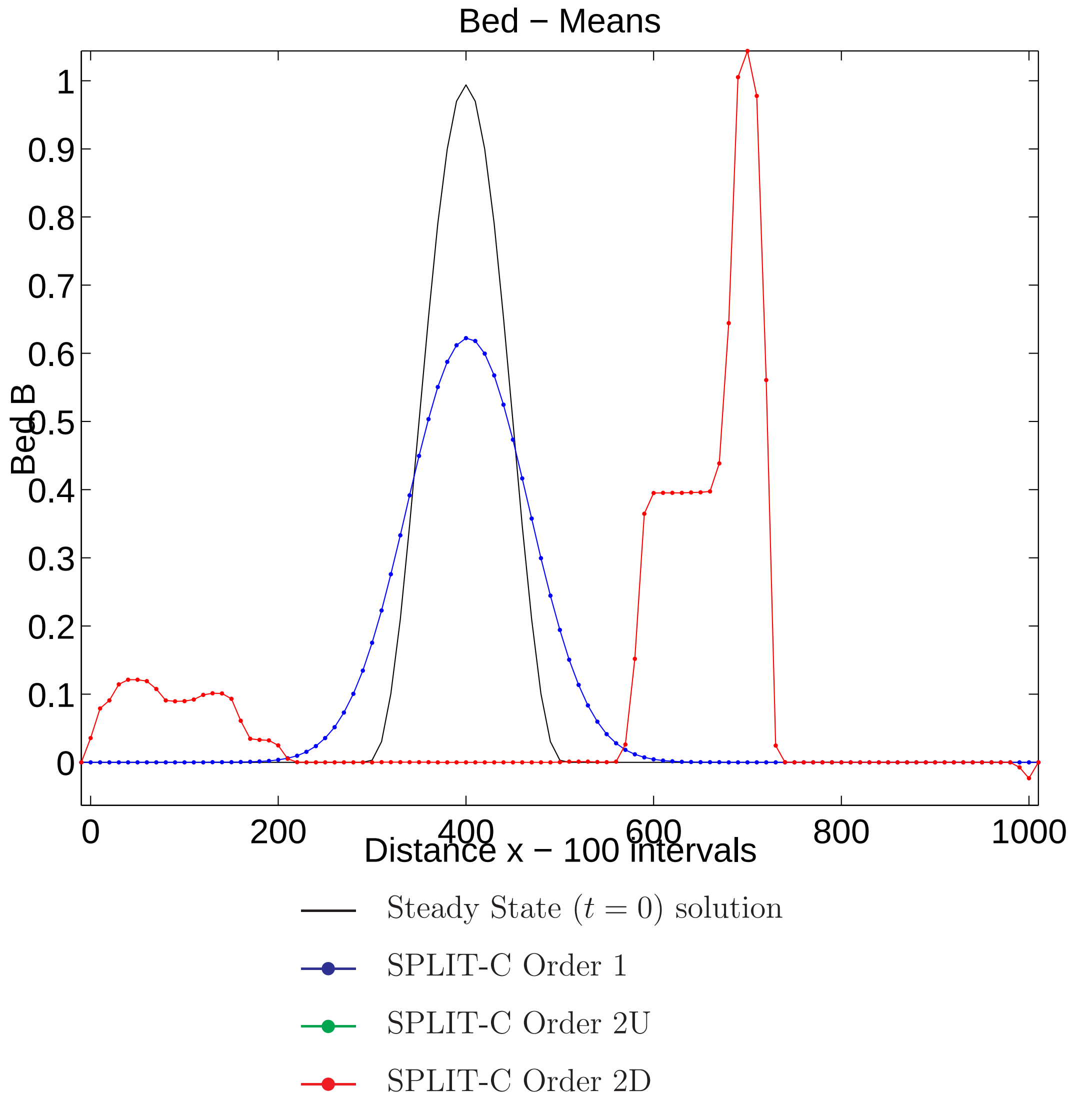


Figure 5.4: Test Case B Results - Standard RKDG with Formulation SPLIT-C Note Order 2U is not shown as it failed to complete, causing a negative depth error.

This assumption required for C-property satisfaction means that the method should not give any accurate results when applied to the morphodynamic equations. This is because, as the bed moves, it has a discontinuous representation. We can already see the effects that the different formulations provide. Surprisingly we have some good results for formulation MORPH-R, we argue that this is because the test case is nice enough to minimise the effect of not satisfying the C-property.

In the next chapter we will provide some advancements to the TVD RKDG method that will improve the results of the method and make it a viable numerical method for the morphodynamical equations.

Chapter 6

Extensions to TVD RKDG

Now that we have defined the TVD RKDG method and shown that the method is not viable for modelling morphodynamics in its current form we can start to modify it. The aim of this process is to generate a method that will give better results than the standard TVD RKDG method. Two areas that we can explore, in search of improvements, are time stepping and the source term discretisation.

6.1 Two Speed Time Stepping

The morphodynamic equations are essentially a two speed flow. As, in practice, the water moves a lot faster in proportion to the bed, typically of the order m/s for water but order mm/s for the bed, the reaction speed of water and bed flow is significantly different. When solving the analytical morphodynamical equations, with a suitable choice of A, we also observe two magnitudes of wave speeds. Ideally we would like the model to be approximated by a method that utilises this difference in "wave speeds". The uncoupled equations, Formulation SPLIT-C allow us to do this.

This uncoupling of the equations gives two systems to solve for, namely the hydrodynamics and the morphodynamics. If these are to be solved at grid spacings proportional to the corresponding wave speeds then we need a system to transfer information from one grid to the other. Let us define the grid spacing for the hydrodynamics to be Δt_w and the grid spacing for the morphodynamics to be Δt_B, with $\Delta t_w << \Delta t_B$ and assume, for simplicity, that the time step for the hydrodynamics is an integer multiple of that of the morphodynamics. This can be seen in Figure 6.1.

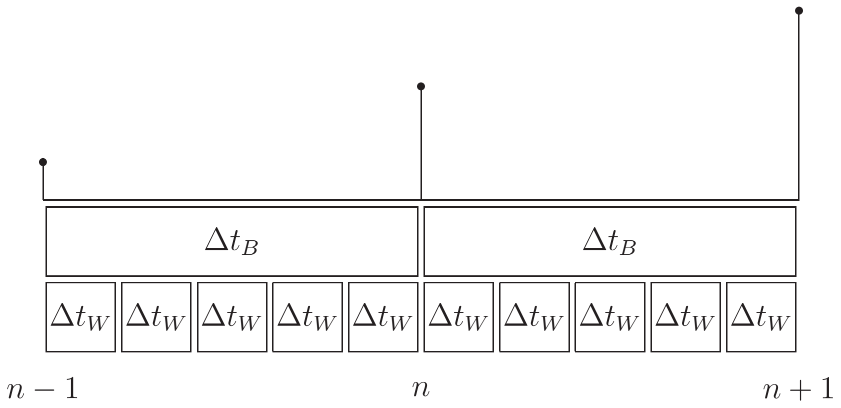


Figure 6.1: Two Speed Time Stepping

Since the bed features in the source term for the hydrodynamics we need to be able to specify the value of the bed at points in time given by $t = i\Delta t_w$. However, the numerical solution to the morphodynamics will define the bed at the points in time given by $t = i\Delta t_B$ but not for all points in time given by $t = i\Delta t_w$. We need some method for interpolating the solution at intervening points. To achieve this we consider modifications to the time stepping algorithms given for the TVD RKDG method. Firstly we will consider forward Euler, since this is the foundation of the TVD RK time stepping, and will then extend this to TVD RK.

6.1.1 Two Speed Forward Euler

With two speed iteration, the temporal grid spacing for the faster flowing water is a lot smaller than that for the slower moving bed. We will, therefore, have many solution points for the water flow between any two solution points for the bed, see Figure 6.1 and at these intermediate points the value of the bed will not be defined. This causes a problem as the water flow at these intermediate points is dependent on the value of the bed.

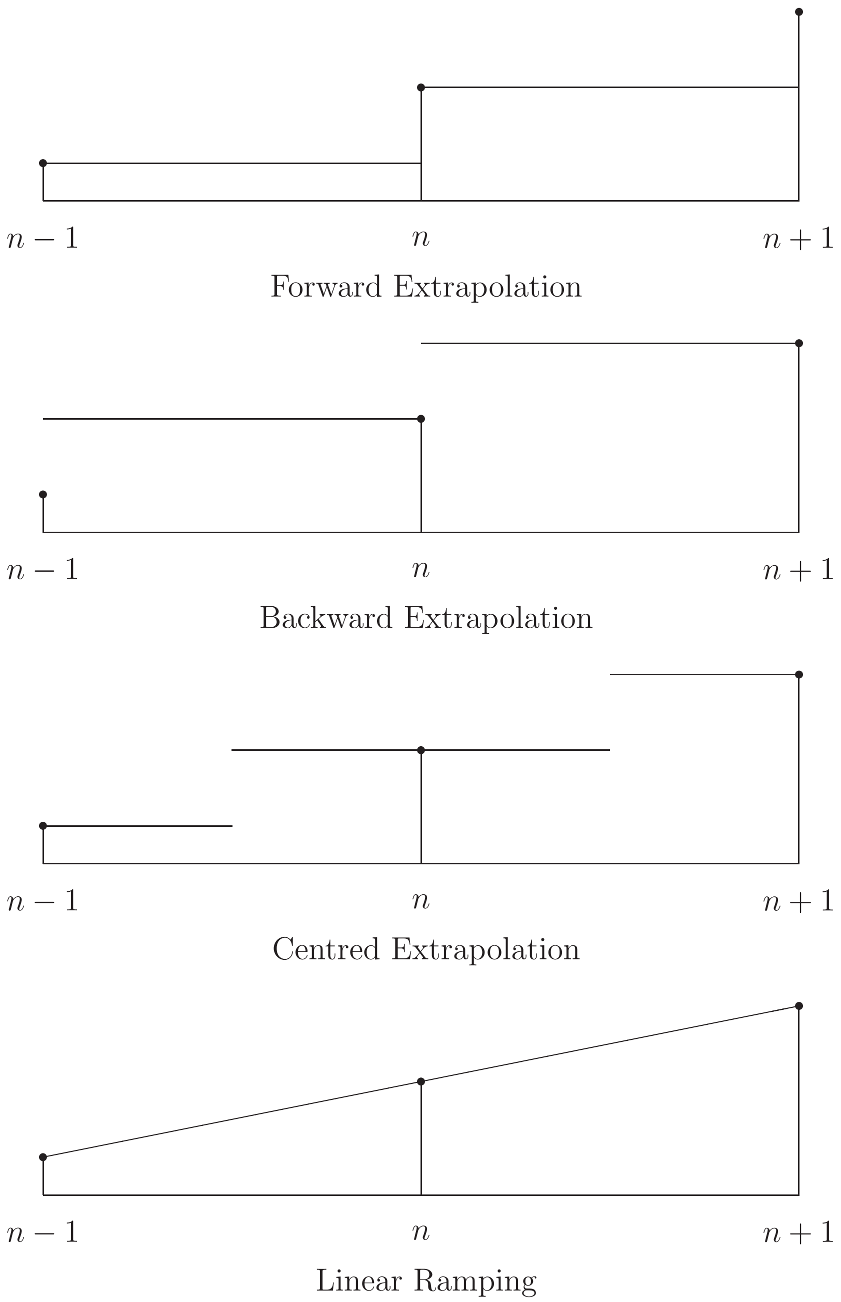
We, therefore, need a definition of the value of the bed at these intermediate points. This definition depends on how the time stepping algorithm is viewed in a continuous manner. Figure 6.2 gives four possible interpretations for the value of the bed at all intermediate values. In this figure, the solution at any intermediate point is determined via extrapolation of the solution from the adjacent time steps.

- Forward Extrapolation - The bed remains where it is and the water works its way forward in time. The bed then makes a leap to catch up with the water. In this way the bed has the solution given at t^n for any time between t^n and t^{n+1}. Forward extrapolation allows no time for the water to react to the bed jump before the solution is recorded.

- Backward Extrapolation - In terms of numerics, at time t^n we can determine the solution to the bed at $n+1$ via forward Euler. This gives us a new solution to the bed. The bed makes a leap forward in time and the water iterates to catch up. In this way the bed has the solution given at t^{n+1} for any time between t^n and t^{n+1}. Backward extrapolation allows the water an entire bed time step to react to the bed jump before the solution is recorded.

- The third method is to centre the extrapolation and assume that the bed will take either solution for half of the time. In this way the bed will have the solution given at t^n for any time between t^n and $\frac{1}{2}(t^n + t^{n+1})$ and will have the solution given at t^{n+1} for any time between $\frac{1}{2}(t^n + t^{n+1})$ and t^{n+1}. The bed will wait while the water moves half a time step, then the bed will make a leap and then the water will continue to finish the iterations. Centred extrapolation allows the water half a time step to react to the bed movement before the solution is recorded.

- The fourth method is to linearly ramp, or interpolate, the bed in the time step. We can determine the solution of the bed at any point in the time step by a linear weighted average of the solutions at the time steps for the bed.



Forward Extrapolation

Backward Extrapolation

Centred Extrapolation

Linear Ramping

Figure 6.2: Approximations for the solution of the bed between iterations.

For Forward Euler, with the exception of linear ramping, these, effectively, only differ in when the solution in any particular time step is observed or sampled. The methods all iterate between a bed jump and a series of water iterations, with only the point at which we observe the solution differing. This sampling of the solution will occur immediately before, immediately after, or half-way between bed jumps and do not have an impact on the actual solution, just the representation of it.

6.1.2 Two Speed Runge-Kutta

In terms of second order Runge-Kutta and morphodynamics this reaction-time difference proves to be a problem. The first stage of Runge-Kutta is a forward Euler time step. This can be interpreted in any of the four ways shown above. The second stage is in some way an implicit time step. The solution at t^{n+1} relies on the solution at t^n and the initial guess at t^{n+1}.

If forward extrapolation, shown above, is used then the initial guess at t^{n+1} has not allowed the water to react to the bed movement and thus is not near a steady or smooth solution. This means that the actual solution at t^{n+1} cannot possibly be near a steady or smooth solution. It is therefore, not a very accurate representation of the solution at this point in time as real water will continuously react to the bed moving.

Backward extrapolation and centred extrapolation are better as the water has time to react to the bed moving and both the initial guess and the actual solution at t^{n+1} will be smoother and closer to the real water solution, this means that we will be performing calculations on solutions that are more reactive to bed movements. The major difference in the implementation of the two methods is that any reaction, by the water to bed change, will have the entire time step to react with the backwards extrapolation whereas it will only have half a time step to react with centred time stepping. This means that any displacement of the water generated by a bed change will be able to disperse over a significantly larger region of the computational domain for backwards extrapolation than if centred extrapolation is used. For this reason backwards extrapolation will be give a smoother water profile and a better solution than centred extrapolation. We shall, therefore, not consider centred extrapolation in the testing of the methods in this thesis.

Linear ramping should prove to be the most accurate of the four options, however the gains in accuracy are offset by the need to interpolate the bed profile for every water time step. This additional computational effort is almost the same as is required to perform the full RKDG scheme on the combined system, as in formulation MORPH-R. It would seem to be more profitable to spend the extra time and gain the additional accuracy by performing actual time steps rather than use linear interpolation. However, since we are seeking to utilise the difference in flow rates, we do not see this as a viable option and will not include it in the testing of methods in this thesis.

In any of these methods there essentially needs to be an uncoupling of the RK algorithm. This can be described, in terms of the second order RK method as:

- Perform the first stage of RK for the bed to get an approximation to the solution at the next bed time step.
- Iterate the water using water time steps with the full RK to produce the first approximation to the solution at the next bed time step for the water. For a Forward Euler method this would be the actual solution at the next time step.
- Perform the second stage of RK for the bed to create a correction for the bed and produce the actual solution of the bed at the next time step.
- Iterate the water again over the bed time step using full RK and water time steps to produce the solution to the water at the next time step.

This means that to perform two speed Runge-Kutta the water needs to be iterated twice over the bed time step. This is an added computational cost but does create a smooth, uniformly second order method.

We will redefine the split formulation to account for the time stepping description. We will define formulation SPLIT-CF to be the split equations given in Section 2.3.4 combined with the forward extrapolation outlined above and formulation SPLIT-CB to be the same equations but with using backward extrapolation.

6.2 Source Term Discretisations

As we saw in Section 5.1, the standard RKDG method, as defined by Cockburn *et al.* [15] and then extended to the shallow water equations by Schwanenberg [68] does not always satisfy the C-property. In the presence of a discontinuous representation of the bed, the solution assumes a steady state that is not physically correct. We would, therefore, like to improve the method to account for this feature.

The natural place to look for the cause of the lack of C-property satisfaction is the source term discretisation. The numerical quadrature applied to the source term does not account for the discontinuities that occur between cells. This is because the numerical quadrature is a function of the interior points only and this is also why we see a lack of balance that is independent of the test function choice. We need a source term discretisation that accounts for the discontinuities at the cell boundaries in addition to the solution inside the cell.

6.2.1 Integration By Parts

A suggestion for the source term discretisation follows the argument that, since the original analytical equations balance, if we apply the same process to the source term as we do to the flux term then we could expect that balance to be retained. This, unfortunately does not help us though as we shall demonstrate.

The major element in the weak formulation for finite elements is the test function and integration by parts. Essentially, the source terms of all of the formulations can be written in the form,

$$\mathbf{R} = \mathbf{e}p\frac{\partial q}{\partial x},$$

where p and q are simple functions of the conserved variables and $\mathbf{e} = [0, 1]^T$ or

$\mathbf{e} = [0, 1, 0]^T$. This means that we are assuming that the equations are in the form,

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F} = \mathbf{e}p\frac{\partial q}{\partial x}.$$

If we follow the same procedure that is used in defining the DG method, then we multiply by a test function, integrate over a cell and then integrate by parts. This means that we achieve a source term that is in the form,

$$\mathbf{e}\left[vpq\right]_{I_j} - \mathbf{e}\int_{I_j}\frac{\partial(vp)}{\partial x}q\;dx = \mathbf{e}\left[vpq\right]_{I_j} - \mathbf{e}\int_{I_j}\frac{\partial v}{\partial x}pq\;dx - \mathbf{e}\int_{I_j}v\frac{\partial p}{\partial x}q\;dx.$$

We can see that we can apply the same process to this source term as is used in the flux function except for the final term. This means that we are accounting for the boundary discontinuities in addition to the solution inside the cell. However, the final term is in exactly the same format as the original source term but with p and q transposed. The process of integrating by parts the source term is equivalent to solving an equation of the form,

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}(\mathbf{F} - \mathbf{e}pq) = -\mathbf{e}\frac{\partial p}{\partial x}q.$$

It is simple to show that the application of this to formulation MORPH-C actually gives us formulation MORPH-R and vice-versa. Since both of these were tested using the standard TVD RKDG method and proven to not be C-property satisfying we can state that the process of integration by parts does not assist us in simply defining a C-property satisfying source term discretisation. A simple finite element approach to the task of source term discretisation does not provide any obvious improvement.

6.2.2 A Finite Difference Discretisation

Examination of the method in its simplest form leads us to a method of defining a suitable source term discretisation. When we look at the first order DG method, coupled with a forward Euler time stepping algorithm, we can see that this is simply a finite difference/finite volume scheme. Bermudez & Vasquez [7] and later Hudson

[34] defined source term discretisations that led to C-property satisfying schemes. This would suggest that a finite difference source term discretisation could balance the finite element discretisation of the flux term and, indeed, for the first order equation this is so.

When the flux term is discretised using the first order Roe-averaged upwind numerical flux we can define the source term discretisation to be,

$$^1\!/_2[\nu(I + |A|A^{-1})\mathbf{R}]_{j-1/2} + {}^1\!/_2[\nu(I - |A|A^{-1})\mathbf{R}]_{j+1/2}, \tag{6.1}$$

where,

$$\mathbf{R}_{j\pm 1/2} = \begin{bmatrix} 0 \\ -g\bar{h}_{j\pm 1/2}\frac{(B_{j\pm 1/2+} - B_{j\pm 1/2+})}{\Delta x} \end{bmatrix}. \tag{6.2}$$

It is clear to see that this reduces to the same finite difference scheme that Bermudez & Vazquez and then Hudson defined. We use the values of $\mathbf{W}_{j\pm 1/2\pm}$ instead of $\mathbf{W}_{j\pm 1}$ since using the values at the cell centres would not make full use of the high order polynomial representation of the solution. As all odd order polynomial basis functions take the value zero at the cell centre the value of the solution at the centre of the cell depends only upon the even order polynomials in its representation. Using the cell centres on a piecewise linear representation would be equivalent to reducing the polynomial representation to piecewise constant, or a first order representation, thus reducing the DG method to first order accurate only.

Similarly we can apply the same process using the second order Roe-averaged Lax-Wendroff flux and define the source term discretisation to be,

$$^1\!/_2[\nu(I + sA)\mathbf{R}]_{j-1/2} + {}^1\!/_2[\nu(I - sA)\mathbf{R}]_{j+1/2}, \tag{6.3}$$

where $s = \Delta t/\Delta x$ and $\mathbf{R}$ is defined by (6.2).

When it comes to higher order DG methods, things are not quite this simple; an assumption that has been used in deriving the finite difference scheme no longer applies in finite elements. To demonstrate this assumption we will consider the Lax-Wendroff finite difference scheme, concentrate on the first order terms and disregard the higher order terms for clarity. The scheme defines the flux discretisation to be,

$$\frac{\mathbf{H}_{j+1/2} - \mathbf{H}_{j-1/2}}{\Delta x},$$

whilst the source term discretisation is defined to be,

$$\tfrac{1}{2}\mathbf{R}_{j+1/2} + \tfrac{1}{2}\mathbf{R}_{j-1/2}.$$

We are, therefore, attempting to balance a difference and an average. Since the numerical flux function, to first order, is defined to be,

$$\mathbf{H}_{j\pm1/2} = \tfrac{1}{2}\mathbf{F}_{j\pm1/2-} + \tfrac{1}{2}\mathbf{F}_{j\pm1/2+},$$

we can expect that the scaling will balance if the definition of $\mathbf{R}$ includes a division by Δx and is in the form of a difference. By defining $\mathbf{R}$ as shown in (6.2) we, indeed, have the division by Δx and it is in the form of a difference. However, the resulting scheme attempts to balance a difference of averages, the flux terms, against an average of differences, the source terms.

For finite differences this works because of the assumption that the solution is represented as piecewise constant and the two interior values are the same, *i.e.* $\mathbf{W}_{j-1/2+} = \mathbf{W}_{j+1/2-}$. This balance can not be achieved by solutions which are represented by linear or higher polynomials in a cell. In terms of the overall scheme, this source term discretisation can account for the discontinuities at the boundaries but not the solution inside the cell. A simple finite difference approach to the task of source term discretisation does not provide any obvious improvement.

6.2.3 The Proposed Discretisation

So far we have, essentially, given two possible source term discretisations that rely on assumptions for satisfaction of the C-property. We will define the FE discretisation to mean using Gaussian quadrature given by (4.13) and the FD discretisation to mean using the discretisation by Bermudez & Vasquez [7] given by the combination of (6.1) and (6.2) or (6.3) and (6.2).

These two discretisations are consistent, in that the error tends to zero as the mesh size is reduced, only when an assumption is made on the solution. The FE discretisation requires that the source term representation is continuous at cell boundaries which translates to a requirement that the bed is continuously represented.

The FD discretisation requires the use of a representation that is piecewise constant. The lack of consistency when these assumptions are not made means that these discretisations will tend to the wrong steady state and cannot satisfy the C-property.

The source term discretisation proposed here is rooted in the observation that these two assumptions and discretisations are complementary. We can see that when the FD assumption is true then the FE discretisation evaluates to zero. Additionally we can see that when the FE assumption is true then the FD discretisation evaluates zero. We propose that a combination of these discretisations could satisfy the C-property. Indeed, the superposition of these source term discretisations actually satisfies the C-property without the need for any of the assumptions given by the separate discretisations, as we shall prove. The FD discretisation accounts for the discontinuities at the boundary of the cells and the FE discretisation accounts for the solution inside the cell.

We therefore define the source term discretisation to be,

$$\begin{aligned}&\frac{\Delta x}{2}\Big[\nu(I+|A|A^{-1})\mathbf{R}\Big]_{j-1/2}+\frac{\Delta x}{2}\Big[\nu(I-|A|A^{-1})\mathbf{R}\Big]_{j+1/2} \\ &+\frac{\Delta x}{2}\left(\nu\mathbf{R}\Big|_{j-1/2\sqrt{3}}+\nu\mathbf{R}\Big|_{j+1/2\sqrt{3}}\right),\end{aligned} \tag{6.4}$$

when the first order Roe-averaged upwind numerical flux function is used and,

$$\begin{aligned}&\frac{\Delta x}{2}\Big[\nu(I+sA)\mathbf{R}\Big]_{j-1/2}+\frac{\Delta x}{2}\Big[\nu(I-sA)\mathbf{R}\Big]_{j+1/2} \\ &+\frac{\Delta x}{2}\left(\nu\mathbf{R}\Big|_{j-1/2\sqrt{3}}+\nu\mathbf{R}\Big|_{j+1/2\sqrt{3}}\right),\end{aligned}$$

when the second order Roe-averaged centred Lax-Wendroff numerical flux function is used. $\mathbf{R}_{j+1/2}$ is defined by (6.2).

6.3 C-property Proof

We will now demonstrate that the DG method, combined with the proposed source term discretisation satisfies the C-property. We shall follow the same process as we did in Section 5.1. We make the same assumptions as before,

$$Q \equiv 0,$$

$$h \equiv D - B,$$

The semi discrete form of the equations, for a second order method is given by,

$$\nu = \nu_{(0)}: \quad \frac{\partial}{\partial t}\mathbf{W}_{(0)} = \begin{array}{l} -\frac{1}{\Delta x}\mathbf{H}(\mathbf{W}_{j+1/2-}, \mathbf{W}_{j+1/2+}) + \frac{1}{\Delta x}\mathbf{H}(\mathbf{W}_{j-1/2-}, \mathbf{W}_{j-1/2+}) \\ +\frac{1}{2}\Big[(I + |A|A^{-1})\mathbf{R}\Big]_{j-1/2} + \frac{1}{2}\Big[(I - |A|A^{-1})\mathbf{R}\Big]_{j+1/2} \\ +\frac{1}{2}\left(\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}}) + \mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}})\right). \end{array}$$

$$\nu = \nu_{(1)}: \quad \frac{\partial}{\partial t}\mathbf{W}_{(1)} = \begin{array}{l} -\frac{3}{\Delta x}\mathbf{H}(\mathbf{W}_{j+1/2-}, \mathbf{W}_{j+1/2+}) - \frac{3}{\Delta x}\mathbf{H}(\mathbf{W}_{j-1/2-}, \mathbf{W}_{j-1/2+}) \\ +\frac{3}{\Delta x}\mathbf{F}(\mathbf{W}_{j-1/2\sqrt{3}}) + \frac{3}{\Delta x}\mathbf{F}(\mathbf{W}_{j+1/2\sqrt{3}}) \\ -\frac{3}{2}\Big[(I + |A|A^{-1})\mathbf{R}\Big]_{j-1/2} + \frac{3}{2}\Big[(I - |A|A^{-1})\mathbf{R}\Big]_{j+1/2} \\ +\frac{\sqrt{3}}{2}\left(-\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}}) + \mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}})\right). \end{array}$$

We will assume that we are using the first order Roe-averaged upwind numerical flux, giving,

$$\mathbf{H}(\mathbf{W}_{L}, \mathbf{W}_{R}) = \frac{1}{2}\begin{bmatrix} -\sqrt{1/2\, g(h_{L} + h_{R})}(h_{R} - h_{L}) \\ 1/2\, g{h_{L}}^2 + 1/2\, g{h_{R}}^2 \end{bmatrix}.$$

Also, for slow flow, $|u| < c$ we have,

$$I + |A|A^{-1} = \frac{1}{c}\begin{bmatrix} c - u & 1 \\ c^2 - u^2 & c + u \end{bmatrix} = \frac{1}{\sqrt{gh}}\begin{bmatrix} \sqrt{gh} - u & 1 \\ gh - u^2 & \sqrt{gh} + u \end{bmatrix},$$

$$I - |A|A^{-1} = \frac{1}{c}\begin{bmatrix} c + u & -1 \\ u^2 - c^2 & c - u \end{bmatrix} = \frac{1}{\sqrt{gh}}\begin{bmatrix} \sqrt{gh} + u & -1 \\ u^2 - gh & \sqrt{gh} - u \end{bmatrix}.$$

Under the C-property assumptions this becomes,

$$I + |A|A^{-1} = \begin{bmatrix} 1 & 1/\sqrt{g\bar{h}} \\ \sqrt{g\bar{h}} & 1 \end{bmatrix}, \qquad I - |A|A^{-1} = \begin{bmatrix} 1 & -1/\sqrt{g\bar{h}} \\ -\sqrt{g\bar{h}} & 1 \end{bmatrix}.$$

This means that,

$$\Big[(I + |A|A^{-1})\mathbf{R}\Big]_{j-1/2} = \begin{bmatrix} -\sqrt{1/2\, g(h_{--} + h_{-+})}(B_{-+} - B_{--})/\Delta x \\ -1/2\, g(h_{--} + h_{-+})(B_{-+} - B_{--})/\Delta x \end{bmatrix},$$

$$\Big[(I-|A|A^{-1})\mathbf{R}\Big]_{j+\frac{1}{2}} = \begin{bmatrix} \sqrt{\tfrac{1}{2}g(h_{+-}+h_{++})}(B_{++}-B_{+-})/\Delta x \\ -\tfrac{1}{2}g(h_{+-}+h_{++})(B_{++}-B_{+-})/\Delta x \end{bmatrix}.$$

The C-property assumptions then give,

$$\Big[(I+|A|A^{-1})\mathbf{R}\Big]_{j-\frac{1}{2}} = \begin{bmatrix} \sqrt{\tfrac{1}{2}g(h_{--}+h_{-+})}(h_{-+}-h_{--})/\Delta x \\ \tfrac{1}{2}g(h_{--}+h_{-+})(h_{-+}-h_{--})/\Delta x \end{bmatrix},$$

$$\Big[(I-|A|A^{-1})\mathbf{R}\Big]_{j+\frac{1}{2}} = \begin{bmatrix} -\sqrt{\tfrac{1}{2}g(h_{+-}+h_{++})}(h_{++}-h_{+-})/\Delta x \\ \tfrac{1}{2}g(h_{+-}+h_{++})(h_{++}-h_{+-})/\Delta x \end{bmatrix}.$$

We can evaluate the source term quadratures to be,

$$\Delta x\left(\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}})+\mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}})\right) = -g\begin{bmatrix} 0 \\ (h_{+-}+h_{-+})(B_{+-}-B_{-+}) \end{bmatrix},$$

$$\frac{\Delta x}{\sqrt{3}}\left(-\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}})+\mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}})\right) = -g\tfrac{1}{3}\begin{bmatrix} 0 \\ (h_{+-}-h_{-+})(B_{+-}-B_{-+}) \end{bmatrix}.$$

The C-property assumptions give,

$$\Delta x\left(\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}})+\mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}})\right) = g\begin{bmatrix} 0 \\ (h_{+-}+h_{-+})(h_{+-}-h_{-+}) \end{bmatrix},$$

$$\frac{\Delta x}{\sqrt{3}}\left(-\mathbf{R}(\mathbf{W}_{j-1/2\sqrt{3}})+\mathbf{R}(\mathbf{W}_{j+1/2\sqrt{3}})\right) = \tfrac{1}{3}g\begin{bmatrix} 0 \\ (h_{+-}-h_{-+})(h_{+-}-h_{-+}) \end{bmatrix}.$$

Additionally we can evaluate the flux term quadrature to be,

$$\mathbf{F}(\mathbf{W}_{j-1/2\sqrt{3}})+\mathbf{F}(\mathbf{W}_{j+1/2\sqrt{3}}) = \tfrac{1}{3}g\begin{bmatrix} 0 \\ {h_{-+}}^2+h_{+-}h_{-+}+{h_{+-}}^2 \end{bmatrix}.$$

Substituting these into the scheme, multiplying the top equation by $2\Delta x$ and the bottom one by $2\Delta x/3$ gives,

$$\nu=\nu_{(0)}:\quad 2\Delta x\frac{\partial}{\partial t}\mathbf{W}_{(0)} = \begin{bmatrix} \sqrt{\tfrac{1}{2}g(h_{+-}+h_{++})}(h_{++}-h_{+-}) \\ -\tfrac{1}{2}g{h_{+-}}^2-\tfrac{1}{2}g{h_{++}}^2 \end{bmatrix} + \begin{bmatrix} -\sqrt{\tfrac{1}{2}g(h_{--}+h_{-+})}(h_{-+}-h_{--}) \\ \tfrac{1}{2}g{h_{--}}^2+\tfrac{1}{2}g{h_{-+}}^2 \end{bmatrix} + \begin{bmatrix} \sqrt{\tfrac{1}{2}g(h_{--}+h_{-+})}(h_{-+}-h_{--}) \\ \tfrac{1}{2}g(h_{--}+h_{-+})(h_{-+}-h_{--}) \end{bmatrix} + \begin{bmatrix} -\sqrt{\tfrac{1}{2}g(h_{+-}+h_{++})}(h_{++}-h_{+-}) \\ \tfrac{1}{2}g(h_{+-}+h_{++})(h_{++}-h_{+-}) \end{bmatrix} + \begin{bmatrix} 0 \\ g(h_{+-}+h_{-+})(h_{+-}-h_{-+}) \end{bmatrix}.$$

$$\nu=\nu_{(1)}:\quad \frac{2\Delta x}{3}\frac{\partial}{\partial t}\mathbf{W}_{(1)} = \begin{bmatrix} \sqrt{\tfrac{1}{2}g(h_{+-}+h_{++})}(h_{++}-h_{+-}) \\ -\tfrac{1}{2}g{h_{+-}}^2-\tfrac{1}{2}g{h_{++}}^2 \end{bmatrix} + \begin{bmatrix} \sqrt{\tfrac{1}{2}g(h_{--}+h_{-+})}(h_{-+}-h_{--}) \\ -\tfrac{1}{2}g{h_{--}}^2-\tfrac{1}{2}g{h_{-+}}^2 \end{bmatrix} + \tfrac{2}{3}g\begin{bmatrix} 0 \\ {h_{-+}}^2+h_{+-}h_{-+}+{h_{+-}}^2 \end{bmatrix} + \begin{bmatrix} -\sqrt{\tfrac{1}{2}g(h_{--}+h_{-+})}(h_{-+}-h_{--}) \\ -\tfrac{1}{2}g(h_{--}+h_{-+})(h_{-+}-h_{--}) \end{bmatrix} + \begin{bmatrix} -\sqrt{\tfrac{1}{2}g(h_{+-}+h_{++})}(h_{++}-h_{+-}) \\ \tfrac{1}{2}g(h_{+-}+h_{++})(h_{++}-h_{+-}) \end{bmatrix} + \tfrac{1}{3}\begin{bmatrix} 0 \\ g(h_{+-}-h_{-+})(h_{+-}-h_{-+}) \end{bmatrix}.$$

It is clear to see that the top rows of these equations are zero as each term has a corresponding cancelling term. This means that $\frac{\partial h}{\partial t}=0$ so we concentrate on the bottom row,

$$\begin{aligned}
\nu = \nu_{(0)}: \quad 2\Delta x \frac{\partial}{\partial t} Q_{(0)} &= -\tfrac{1}{2} g {h_{+-}}^2 - \tfrac{1}{2} g {h_{++}}^2 \\
&\quad + \tfrac{1}{2} g {h_{--}}^2 + \tfrac{1}{2} g {h_{-+}}^2 \\
&\quad + \tfrac{1}{2} g (h_{--} + h_{-+})(h_{-+} - h_{--}) \\
&\quad + \tfrac{1}{2} g (h_{+-} + h_{++})(h_{++} - h_{+-}) \\
&\quad + g (h_{+-} + h_{-+})(h_{+-} - h_{-+}),
\end{aligned}$$

$$\begin{aligned}
\nu = \nu_{(1)}: \quad \frac{2\Delta x}{3} \frac{\partial}{\partial t} Q_{(1)} &= -\tfrac{1}{2} g {h_{+-}}^2 - \tfrac{1}{2} g {h_{++}}^2 \\
&\quad - \tfrac{1}{2} g {h_{--}}^2 - \tfrac{1}{2} g {h_{-+}}^2 \\
&\quad + \tfrac{2}{3} g {h_{-+}}^2 + \tfrac{2}{3} g h_{+-} h_{-+} + \tfrac{2}{3} g {h_{+-}}^2 \\
&\quad - \tfrac{1}{2} g (h_{--} + h_{-+})(h_{-+} - h_{--}) \\
&\quad + \tfrac{1}{2} g (h_{+-} + h_{++})(h_{++} - h_{+-}) \\
&\quad + \tfrac{1}{3} g (h_{+-} - h_{-+})(h_{+-} - h_{-+}).
\end{aligned}$$

We can expand the double bracket products to get,

$$\begin{aligned}
\nu = \nu_{(0)}: \quad 2\Delta x \frac{\partial}{\partial t} Q_{(0)} &= -\tfrac{1}{2} g {h_{+-}}^2 - \tfrac{1}{2} g {h_{++}}^2 \\
&\quad + \tfrac{1}{2} g {h_{--}}^2 + \tfrac{1}{2} g {h_{-+}}^2 \\
&\quad + \tfrac{1}{2} g {h_{-+}}^2 - \tfrac{1}{2} g {h_{--}}^2 \\
&\quad + \tfrac{1}{2} g {h_{++}}^2 - \tfrac{1}{2} g {h_{+-}}^2 \\
&\quad + g {h_{+-}}^2 - g {h_{-+}}^2,
\end{aligned}$$

$$\begin{aligned}
\nu = \nu_{(1)}: \quad \frac{2\Delta x}{3} \frac{\partial}{\partial t} Q_{(1)} &= -\tfrac{1}{2} g {h_{+-}}^2 - \tfrac{1}{2} g {h_{++}}^2 \\
&\quad - \tfrac{1}{2} g {h_{--}}^2 - \tfrac{1}{2} g {h_{-+}}^2 \\
&\quad + \tfrac{2}{3} g {h_{-+}}^2 + \tfrac{2}{3} g h_{+-} h_{-+} + \tfrac{2}{3} g {h_{+-}}^2 \\
&\quad - \tfrac{1}{2} g {h_{-+}}^2 + \tfrac{1}{2} g {h_{--}}^2 \\
&\quad + \tfrac{1}{2} g {h_{++}}^2 - \tfrac{1}{2} g {h_{+-}}^2 \\
&\quad + \tfrac{1}{3} g {h_{+-}}^2 - \tfrac{2}{3} g h_{-+} h_{+-} + \tfrac{1}{3} g {h_{-+}}^2,
\end{aligned}$$

or, arranged for clarity,

$$
\begin{array}{llll}
\nu = \nu_{(0)}: & 2\Delta x \frac{\partial}{\partial t} Q_{(0)} = & & -\tfrac{1}{2}g{h_{+-}}^2 - \tfrac{1}{2}g{h_{++}}^2 \\
& +\tfrac{1}{2}g{h_{--}}^2 + \tfrac{1}{2}g{h_{-+}}^2 & & \\
& -\tfrac{1}{2}g{h_{--}}^2 + \tfrac{1}{2}g{h_{-+}}^2 & & \\
& & & -\tfrac{1}{2}g{h_{+-}}^2 + \tfrac{1}{2}g{h_{++}}^2 \\
& & -g{h_{-+}}^2 & +g{h_{+-}}^2,
\end{array}
$$

$$
\begin{array}{llll}
\nu = \nu_{(1)}: & \frac{2\Delta x}{3} \frac{\partial}{\partial t} Q_{(1)} = & & -\tfrac{1}{2}g{h_{+-}}^2 - \tfrac{1}{2}g{h_{++}}^2 \\
& -\tfrac{1}{2}g{h_{--}}^2 - \tfrac{1}{2}g{h_{-+}}^2 & & \\
& & +\tfrac{2}{3}g{h_{-+}}^2 + \tfrac{2}{3}gh_{+-}h_{-+} + \tfrac{2}{3}g{h_{+-}}^2 & \\
& +\tfrac{1}{2}g{h_{--}}^2 - \tfrac{1}{2}g{h_{-+}}^2 & & \\
& & & -\tfrac{1}{2}g{h_{+-}}^2 + \tfrac{1}{2}g{h_{++}}^2 \\
& & +\tfrac{1}{3}g{h_{-+}}^2 - \tfrac{2}{3}gh_{-+}h_{+-} + \tfrac{1}{3}g{h_{+-}}^2. &
\end{array}
$$

Therefore the method satisfies the C-property with no restrictions upon the solution. For a first order method we need to simply consider the first of the two equations given in the proof above.

6.4 Results

To validate the method, when combined with the proposed source term discretisation and the two speed time stepping, we use the same tests as we did for the standard RKDG method. We will show the results for when the FE discretisation, the FD discretisation and the proposed discretisation are used. Reference should be made to Section 5.3 for information on how to interpret the results.

6.4.1 Test Case A

Figure 5.1 shows the results for test problem A when the source term is discretised using the standard approach, FE discretisation. Since the solution is a steady state the time stepping is irrelevant and so formulations SPLIT-CF and SPLIT-CB are identical. These results have already been discussed in Section 5.3.

Figure 6.3 shows the results for test problem A where the source term has been discretised using the finite difference approach given by (6.1). For all formulations the first order methods give the exact answer, as expected. Additionally, for all orders and all formulations it is clear that the step on the right hand side of the domain is approximated correctly. This can be expected as the entire change in the bed profile within this step is achieved via the discontinuity.

For all of the second order limited and unlimited methods we have a change in the steady state on the smoothly represented bump. This is consistent with the fact that the source term discretisation does not account for the solution within the cell. The limited and unlimited methods differ by machine precision only. We see a difference with formulation MORPH-R as compared to the other formulations as it is partially aware of the bed profile through the flux term, however, since it does not satisfy the C-property we do not have the correct steady state.

Figure 6.4 shows the results for test problem A where the source term has been approximated using the proposed source term discretisation given by (6.4). It is clear that all formulations and all orders of methods produce results that are within machine precision of the exact solution. This is in agreement with the proof that the method satisfies the C-property.

Figure 6.5, Figure 6.6 and Figure 6.7 give results for Test A when a non-uniform grid is used. The grid is generated with cell j having width,

$$\Delta x_j = 1000 \frac{{}^1\!/\!_2 \left(3 + \sin\left(\frac{2\Pi j}{J}\right)\right)}{\sum_{j=1}^{J} {}^1\!/\!_2 \left(3 + \sin\left(\frac{2\Pi j}{J}\right)\right)},$$

where J is the number of cells in the domain. The denominator of this equation, along with the initial thousand, is used as a normalising term to ensure that the last cell is centred on the point $x = 1000$. This gives a smoothly varying cell width for which the ratio of largest to smallest cell is 2. We can see that the features observed in the uniform grid are also apparent in the non-uniform grid.

There is a new feature of a long oscillation around the bump for the FE discretisation for which no explanation is currently available. The proposed source term discretisation with any of the formulations has an error within machine precision of the analytical solution. This clearly demonstrates that the proposed discretisation satisfies the C-property even on non-uniform grids.

6.4.2 Test Case B

Figure 6.8 to Figure 6.19 give the results for all orders and all formulations applied to Test Case B with the three possible source term discretisations. We have displayed the region $\{x : 400 \leq x \leq 900\}$ to improve the visibility of the results. We will comment on results for each type of source term discretisation separately.

FE Source Term Discretisation

We have already proved that the FE source term discretisation fails to satisfy the C-property. We should, therefore, not expect to see good results for water flow when this discretisation is used and, indeed, Figure 6.8 to Figure 6.11 show poor water flow results.

For formulation MORPH-C, Figure 6.8, we have very poor bed results. All orders fail to move the solution at, anywhere near, the correct speed. The second order unlimited method fails to produce any results at all. The second order limited method does achieve results but, whilst attempting to eliminate the oscillations that caused the unlimited method to blow up, it causes the bed profile to “square up”.

For formulation MORPH-R, Figure 6.9, we achieve some fairly good results for the bed profile, displaying the classic characteristics of the orders, however the water profile is poor. Due to the deep water depth, the bed results are not heavily polluted by the poor water flow. In this case, although the second order unlimited method creates large oscillations in the water flow, they are maintained within a stable region for the duration of this test. Using this scheme it is possible to generate negative depths, via the oscillations, in finite time by using a test case where the depth is

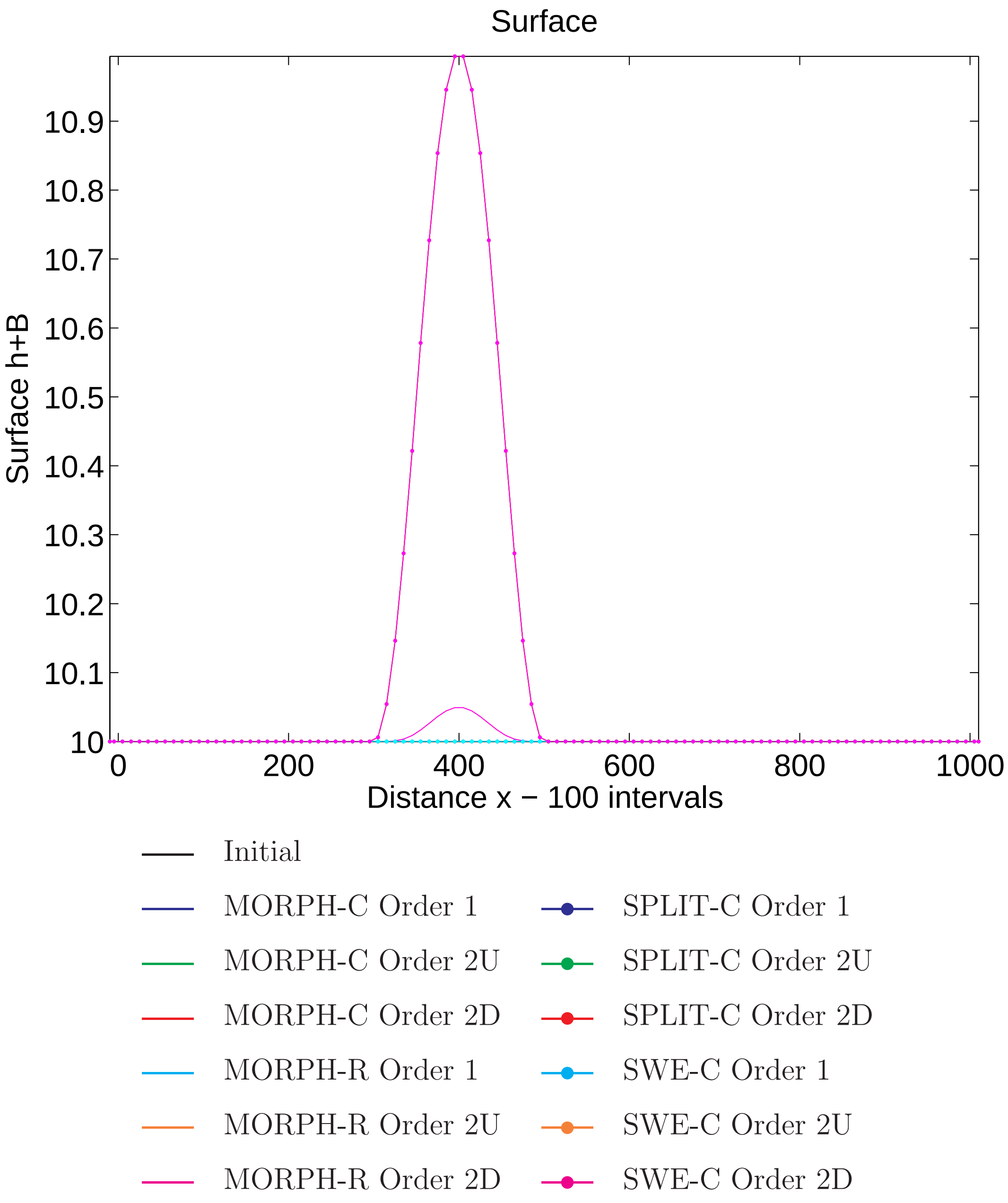


Figure 6.3: Test Case A Results with FD Source Discretisation

All Order 1 results are coincident and are represented by the cyan dotted line. MORPH-R Order 2U and Order 2D are coincident and are represented by the magenta line. SWE-C, SPLIT-C and MORPH-C, Order 2U and Order 2D are all coincident and are represented by the magenta dotted line.

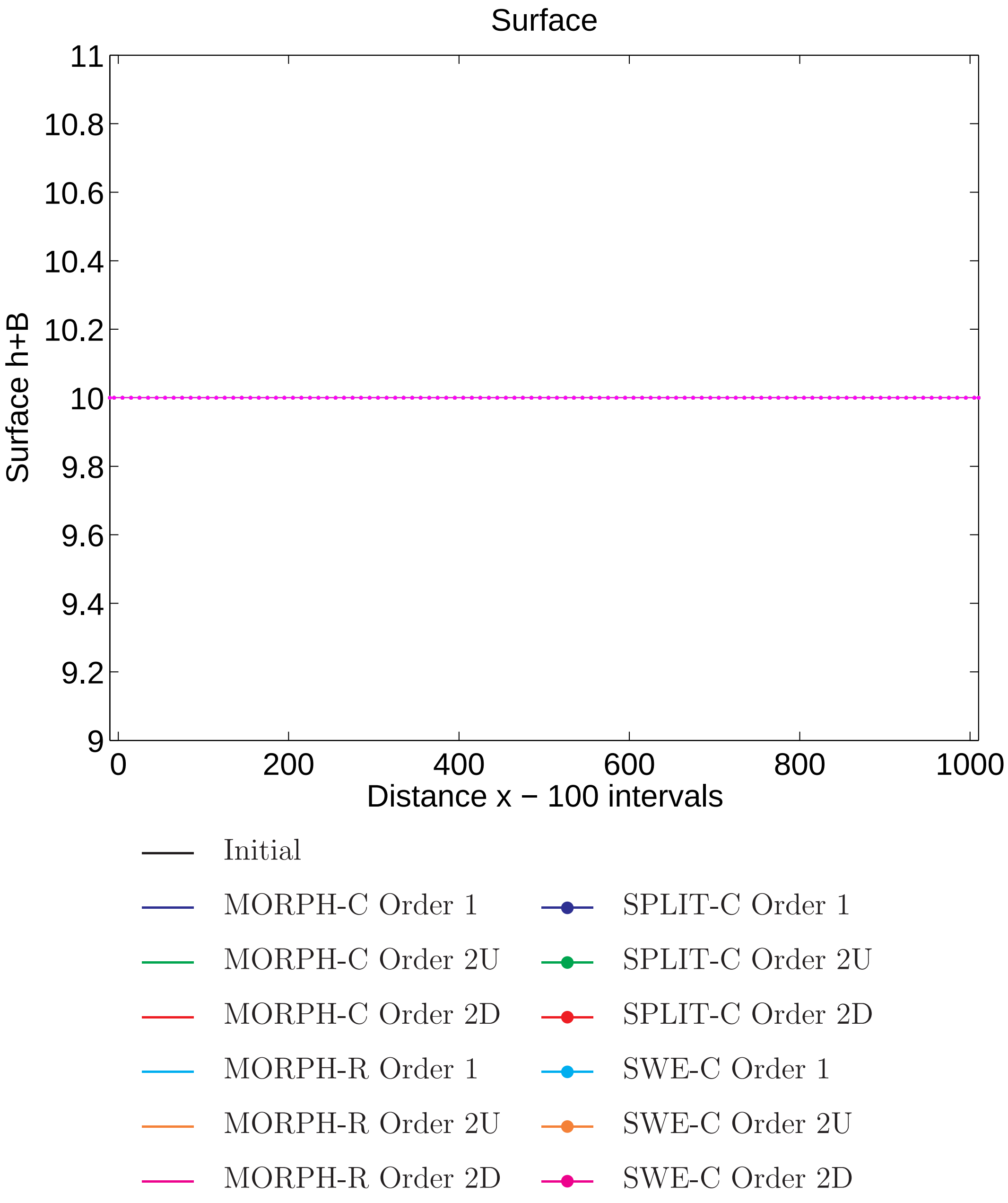


Figure 6.4: Test Case A Results with Proposed Source Discretisation

All results are within machine precision of the analytical solution.

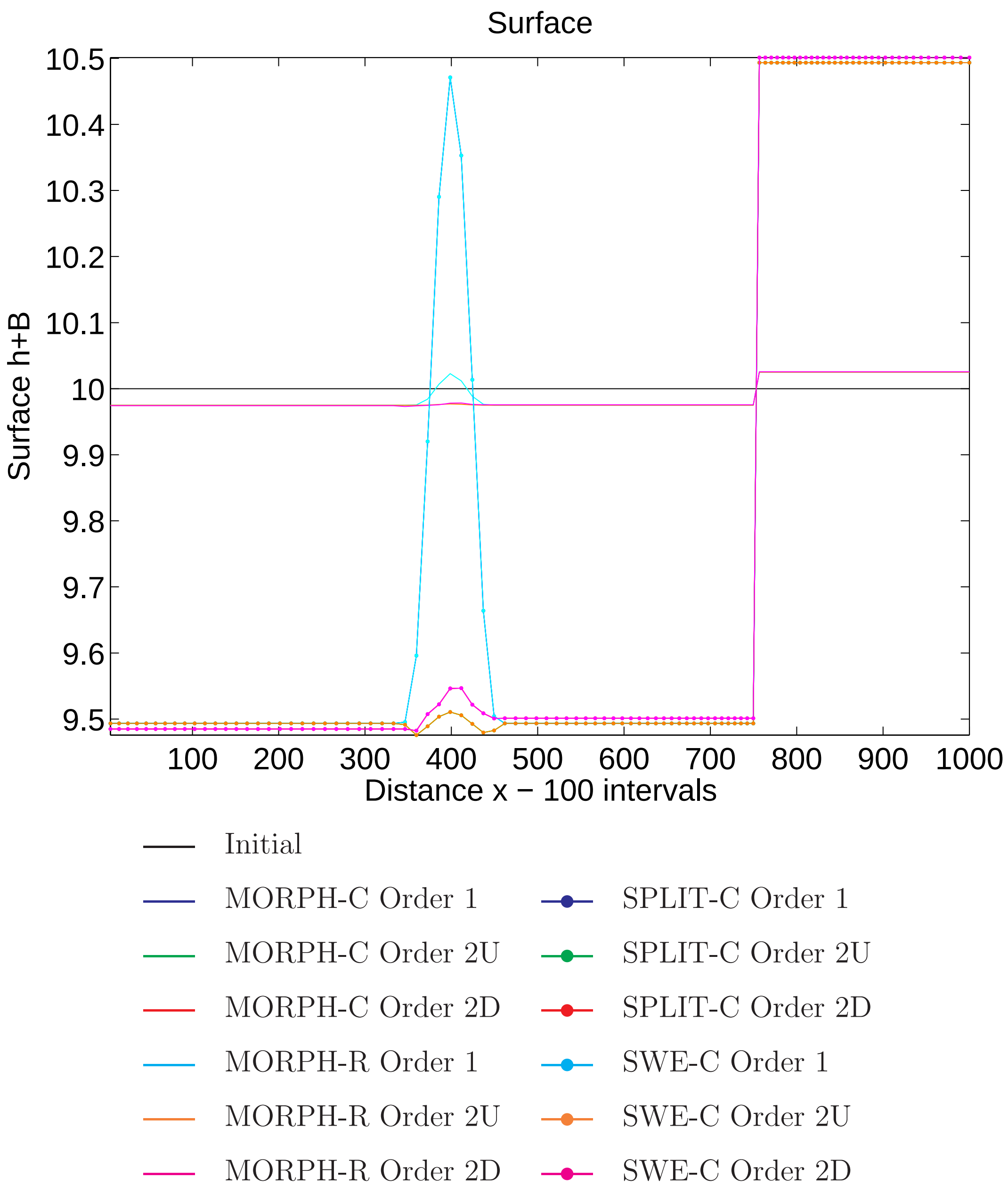


Figure 6.5: Test Case A Results with FE Source Discretisation On A Non-uniform Grid

For MORPH-C, SWE-C and SPLIT-C, Order 1 are coincident and represented by the cyan dotted line, Order 2U are coincident and represented by the orange dotted line and Order 2D are coincident and represented by the magenta dotted line. MORPH-R Order 2U and Order 2D are similar.

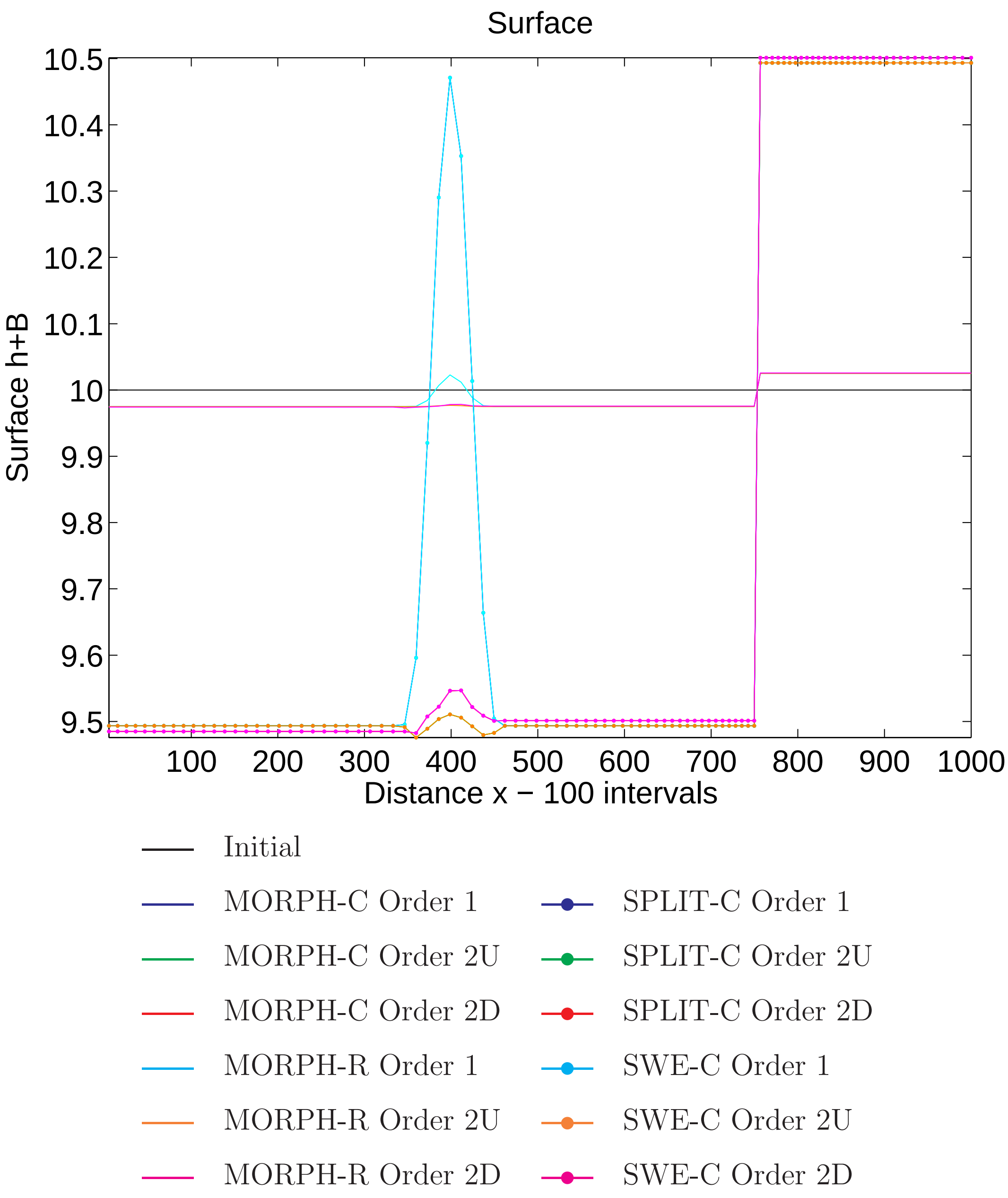


Figure 6.6: Test Case A Results with FD Source Discretisation On A Non-uniform Grid

All Order 1 formulations are identical and are represented by the cyan dotted line. MORPH-C, SWE-C and SPLIT-C, Order 2U are coincident and represented by the orange dotted line and Order 2D are coincident and represented by the magenta dotted line. MORPH-R Order 2U and Order 2D are similar.

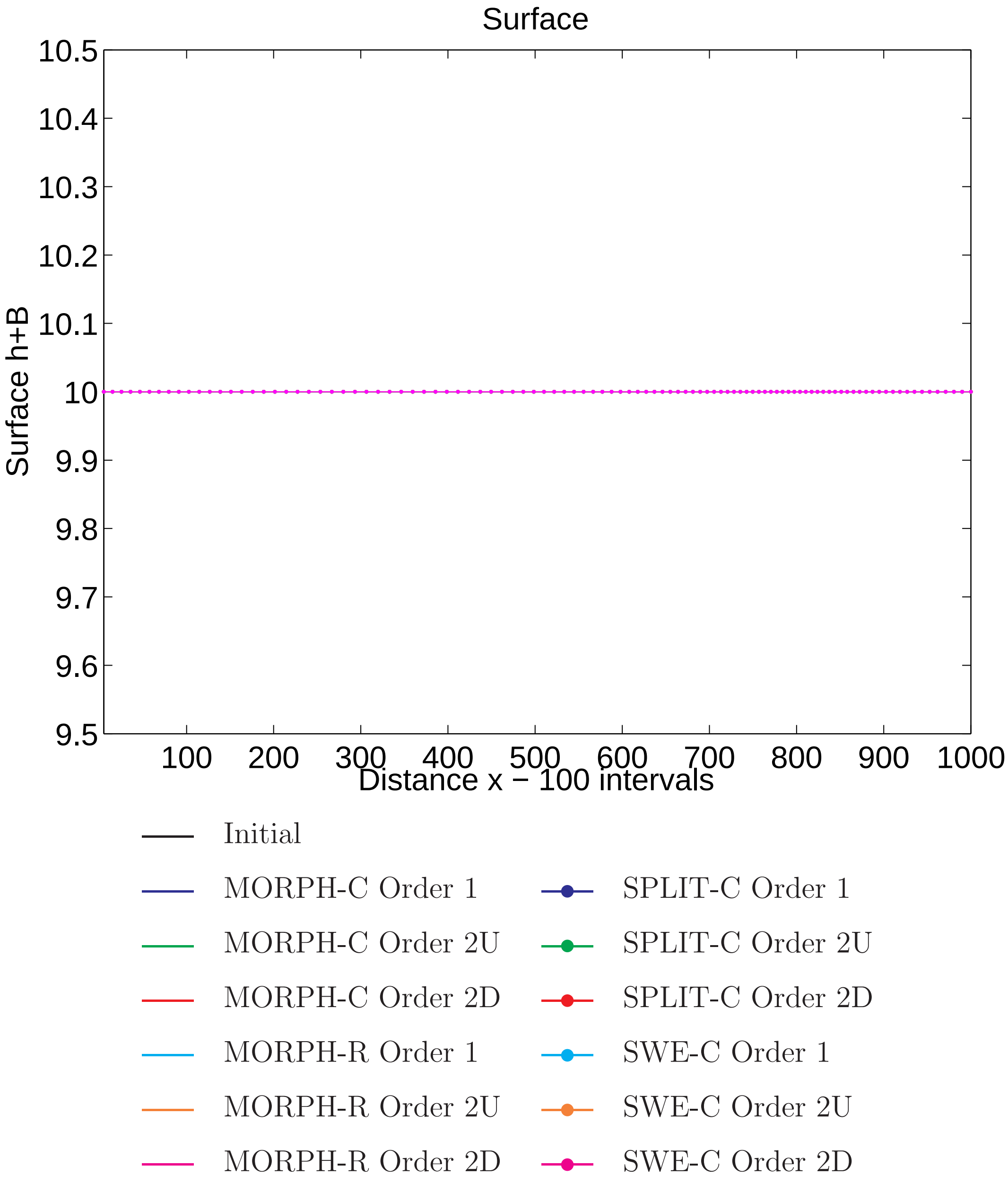


Figure 6.7: Test Case A Results with Proposed Source Discretisation On A Non-uniform Grid

All results are within machine precision of the analytical solution.

small.

For formulation SPLIT-CF, Figure 6.10, we fail to produce second order unlimited results. The second order limited method manages to minimise the oscillations enough to provide a result but the combination of oscillating water and the stepping effect seen with formulation MORPH-C creates the staircase effect on the bed profile. Additionally, the first order method fails to move the bed at all, only diffusing the bed profile where it is.

Formulation SPLIT-CB, Figure 6.11, produces similar but smoother results to formulation SPLIT-CF. This suggests that the lack of C-property satisfaction has a greater effect on the solution than the time stepping method. We, again, have no results for the second order unlimited method and a staircasing effect for the second order limited method.

FD Source Term Discretisation

We know that the FD source term discretisation also fails to satisfy the C-property when the solution is represented by linear or higher polynomials. We can see this in Figure 6.12 to Figure 6.15. For each formulation the first order results show a good water flow and poor results for second order unlimited and limited methods. In the case of formulation SPLIT-CF we have poor results due to the forward extrapolation.

For formulation MORPH-C, Figure 6.12, we see an unusual effect with the first order results. Although the bed has moved at approximately the correct speed it is generating oscillations that are travelling backwards down the bump. These oscillations are not caused by any high order representation as this is a completely first order accurate method.

For the second order limited and unlimited methods the bed profile has hardly changed. Due to the FD discretisation the water flow only sees the bed through its discontinuities. Since the bed is, initially, smoothly represented the water flow, initially, sees no bed. This creates a steady state where the velocity at any point in the domain is constant, not the discharge. Since the bed flux is only dependent on this velocity we have a constant bed flux at every point in the domain. Therefore, although the bed is actually flowing, the bump is moving without deformation over the bed. As numerical error introduces discontinuities the water flow slowly becomes aware of the bed, causing changes in the velocities and thus the flow of the bed.

For formulation MORPH-R, Figure 6.13, the water flow is partially aware of the bed profile through the flux term. It is for this reason that the bed initially moves and we have fairly good results. In the same manner as with the FE source term discretisation we see the classic characteristics of the orders in the bed profile. We still have inaccurate results for the water flow with the second order limited and unlimited methods.

For formulation SPLIT-CF, Figure 6.14, and formulation SPLIT-CB, Figure 6.15, we achieve no results for the second order limited and unlimited methods. The lack of C-property satisfaction for higher order methods create oscillations that are too large to maintain stability. The first order methods generate results that show the classic first order diffusion with the backwards extrapolation generating smoother results. This, again, highlights that the time stepping method is less significant than C-property satisfaction.

Proposed Source Term Discretisation

We have proved that the proposed source term discretisation leads to a C-property satisfying scheme. The effect of this C-property satisfaction provides smooth and accurate results in morphodynamics, as shown by Figure 6.16, Figure 6.17 and Figure 6.19 where the water flow balances the bed profile, even when the bed profile is oscillatory. The apparent lack of smoothness in Figure 6.18 is actually the effect of the forwards extrapolation time stepping, not a lack of C-property satisfaction.

For formulation MORPH-C, Figure 6.16, we see the same trailing oscillations that were present with the FD source term discretisation. These are too large to maintain stability in the second order limited method and are not removed by limiting. We propose that this is actually an effect caused by the non-invertible

Jacobian and is also present in the work by Hudson, seen in formulation A-AF in Figure 3.10 and Figure 3.20 of [34], pages 92 and 102.

For formulation MORPH-R, Figure 6.17, we have excellent results, showing the classic characteristics of the orders, this time with an appropriate water flow.

For formulation SPLIT-CF, Figure 6.18, we have a complete set of results but the interaction of the water and bed is causing excessive oscillations in the water profile for the second order method. It is clear that the forward extrapolation time stepping is the cause of this severe error as the first order results do not show these oscillations. The apparent lack of C-property satisfaction with the first order method is due to the fact that the water flow has not had time to react to the concurrent bed time step. This is due to the water, during the prior time step, settling to the bed profile at the start of that time step, not the profile at the end of the step.

For Formulation SPLIT-CB, Figure 6.19, we have excellent results, showing the classic characteristics of the orders of the method, demonstrating an appropriate water flow.

6.4.3 Comparison of Methods

It is clear that the need to satisfy the C-property is important. The effects of lack of C-property satisfaction range from minor variations on the surface to a coupled resonation between the hydrodynamics and morphodynamics leading to blow up. None of the results using the FD or FE approach alone have produced good water flow results. Due to the choice of test problem, the scale of error in the water flow is of the order 10^{-2}. It is clear that, with some combinations of formulations and source term discretisations, this relatively small inaccuracy in the water profile, can create vastly different results in the bed profile.

For this test case, we can see that formulation MORPH-R is better than formulation MORPH-C. In the cases where formulation MORPH-C could not provide any results, formulation MORPH-R could, not only provide results, but provide good results. The effect of differentiating the equation of momentum by parts ap-

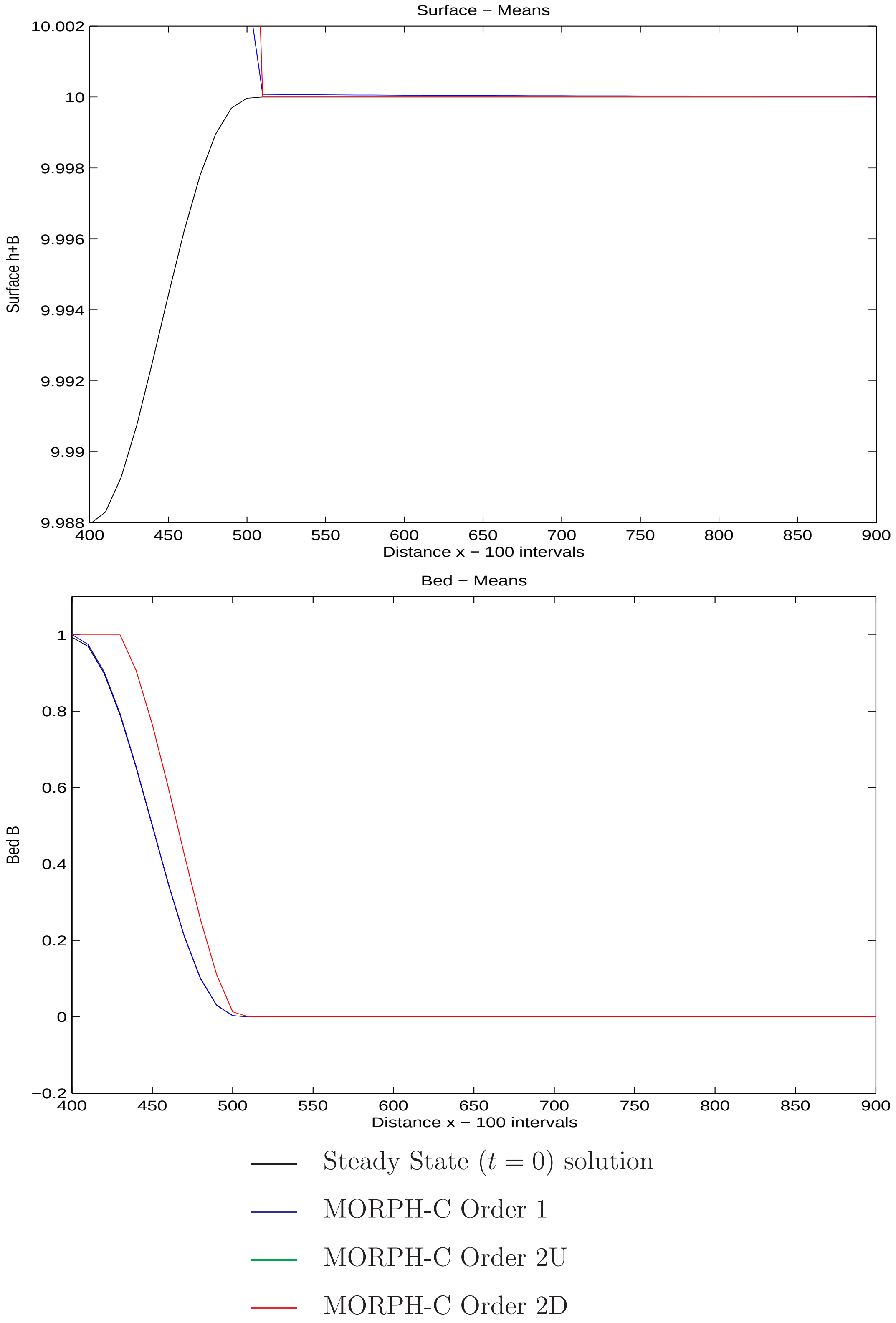


Figure 6.8: Test Case B Results, FE Source Discretisation - MORPH-C

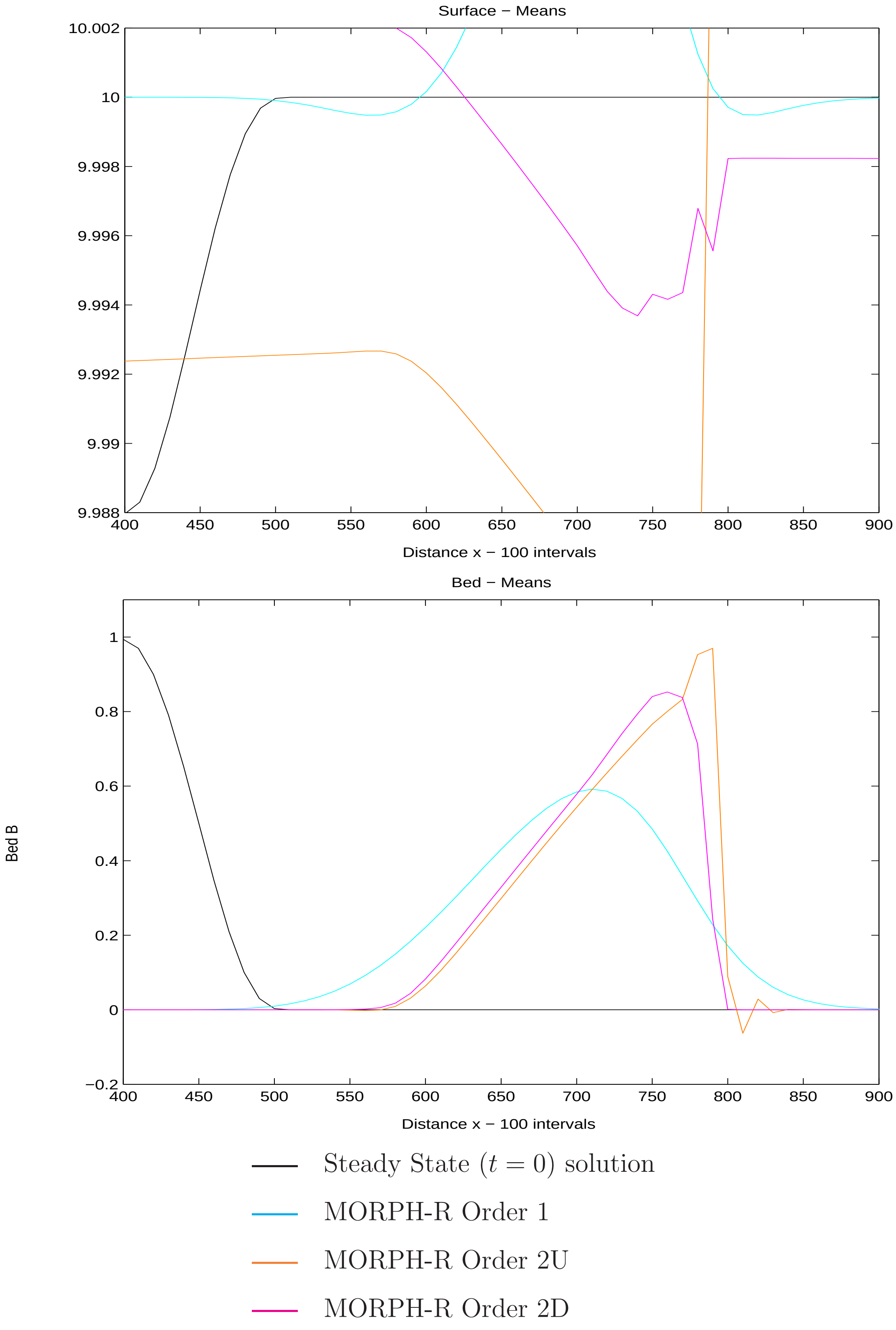


Figure 6.9: Test Case B Results, FE Source Discretisation - MORPH-R

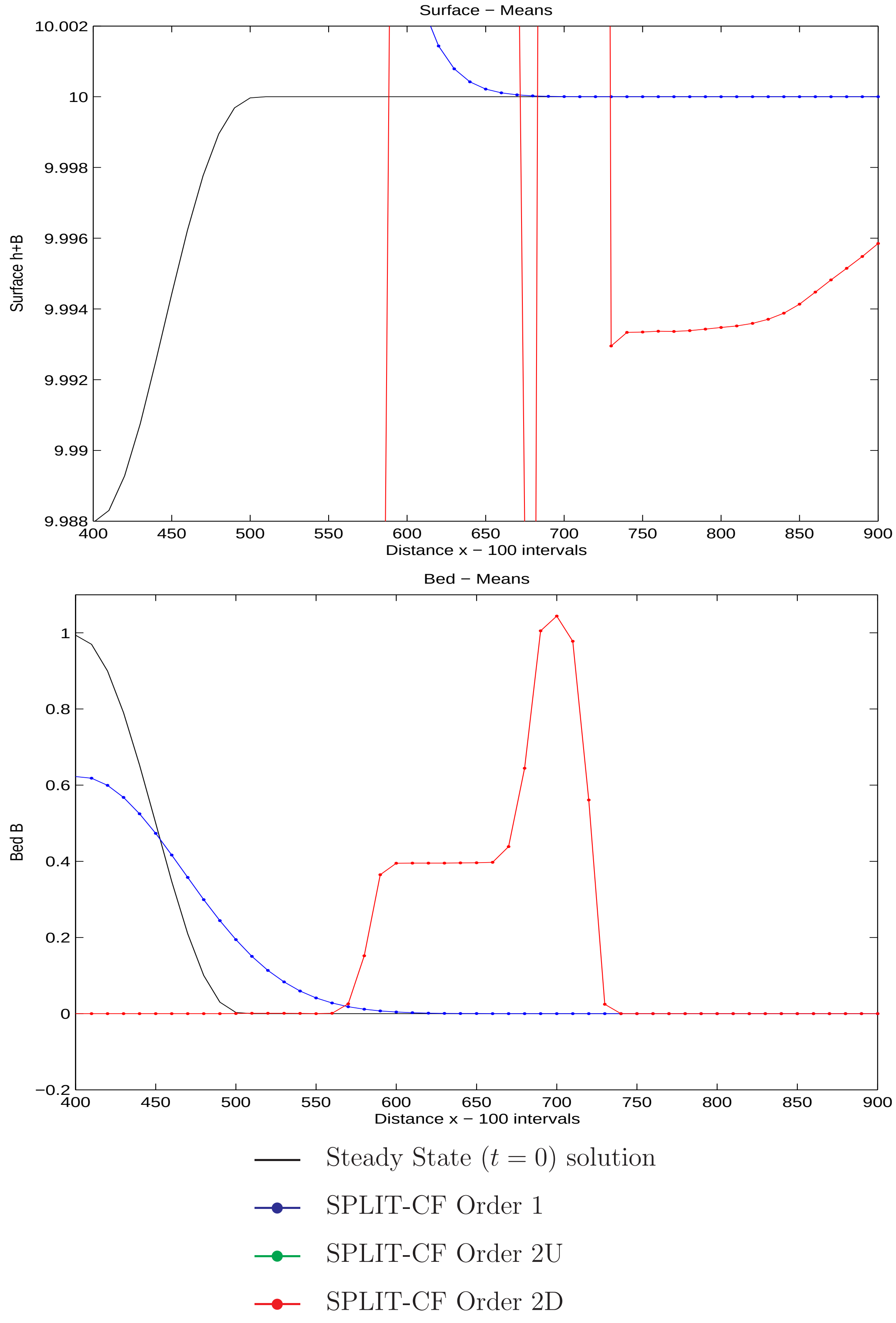


Figure 6.10: Test Case B Results, FE Source Discretisation - SPLIT-CF

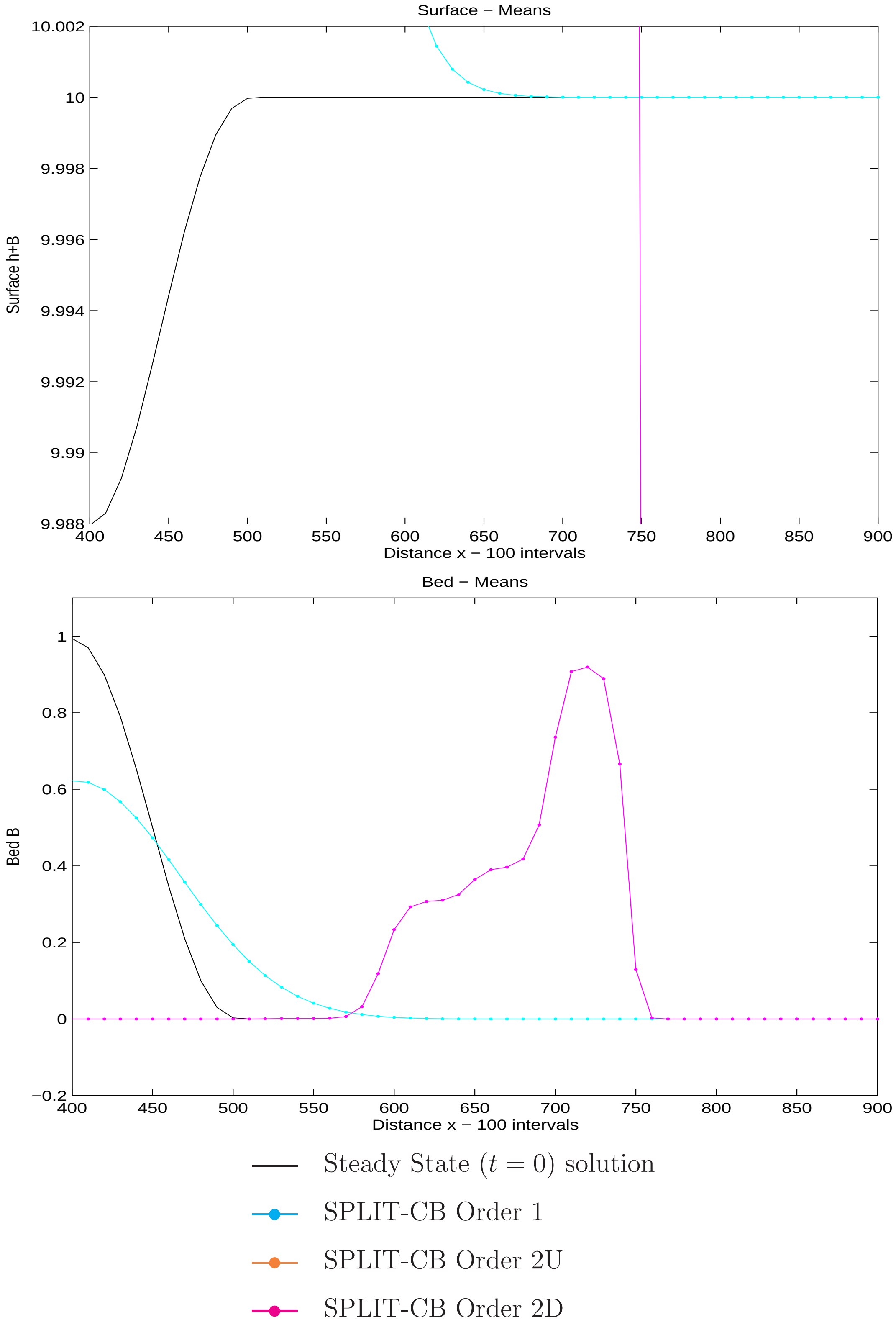


Figure 6.11: Test Case B Results, FE Source Discretisation - SPLIT-CB

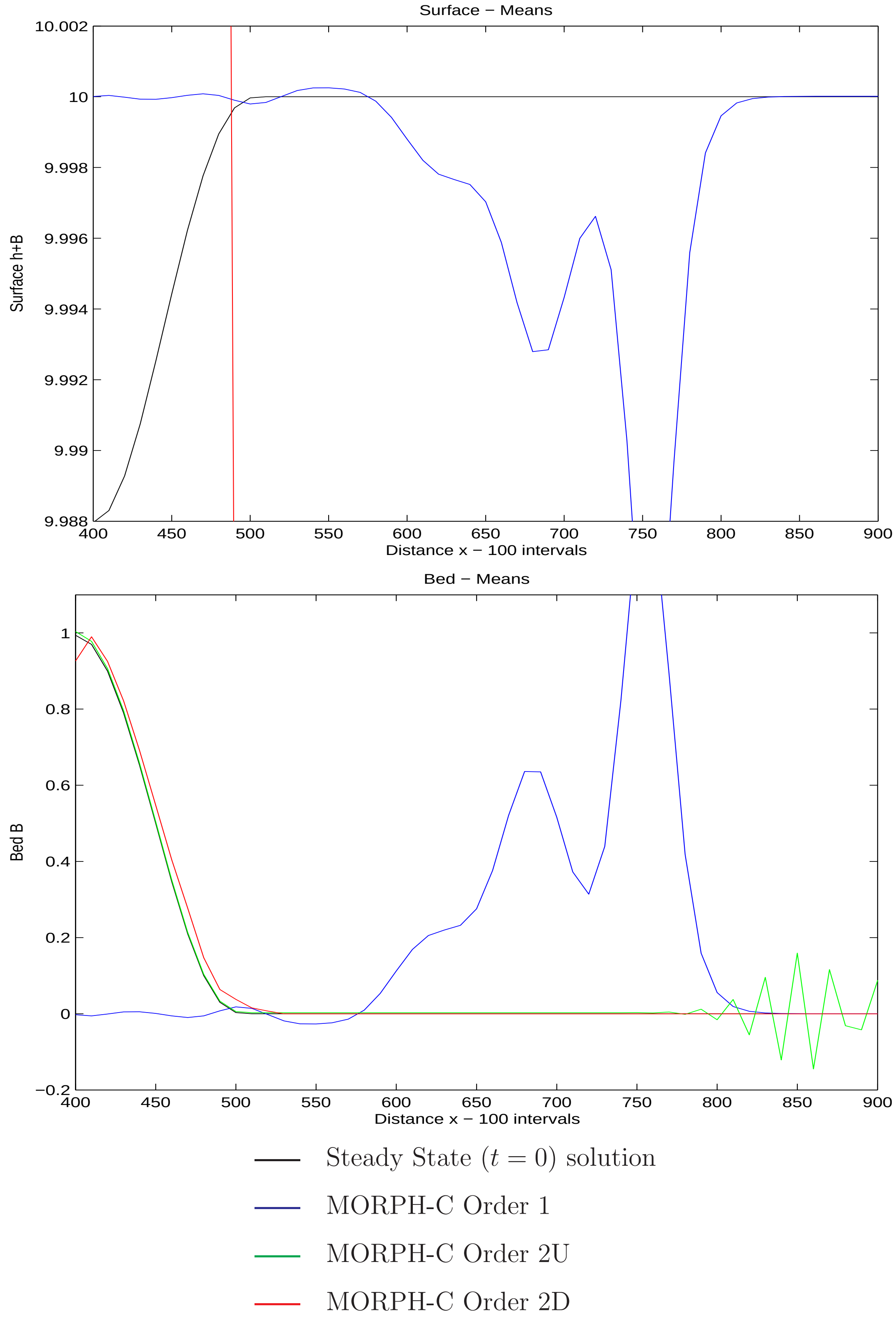


Figure 6.12: Test Case B Results, FD Source Discretisation - MORPH-C

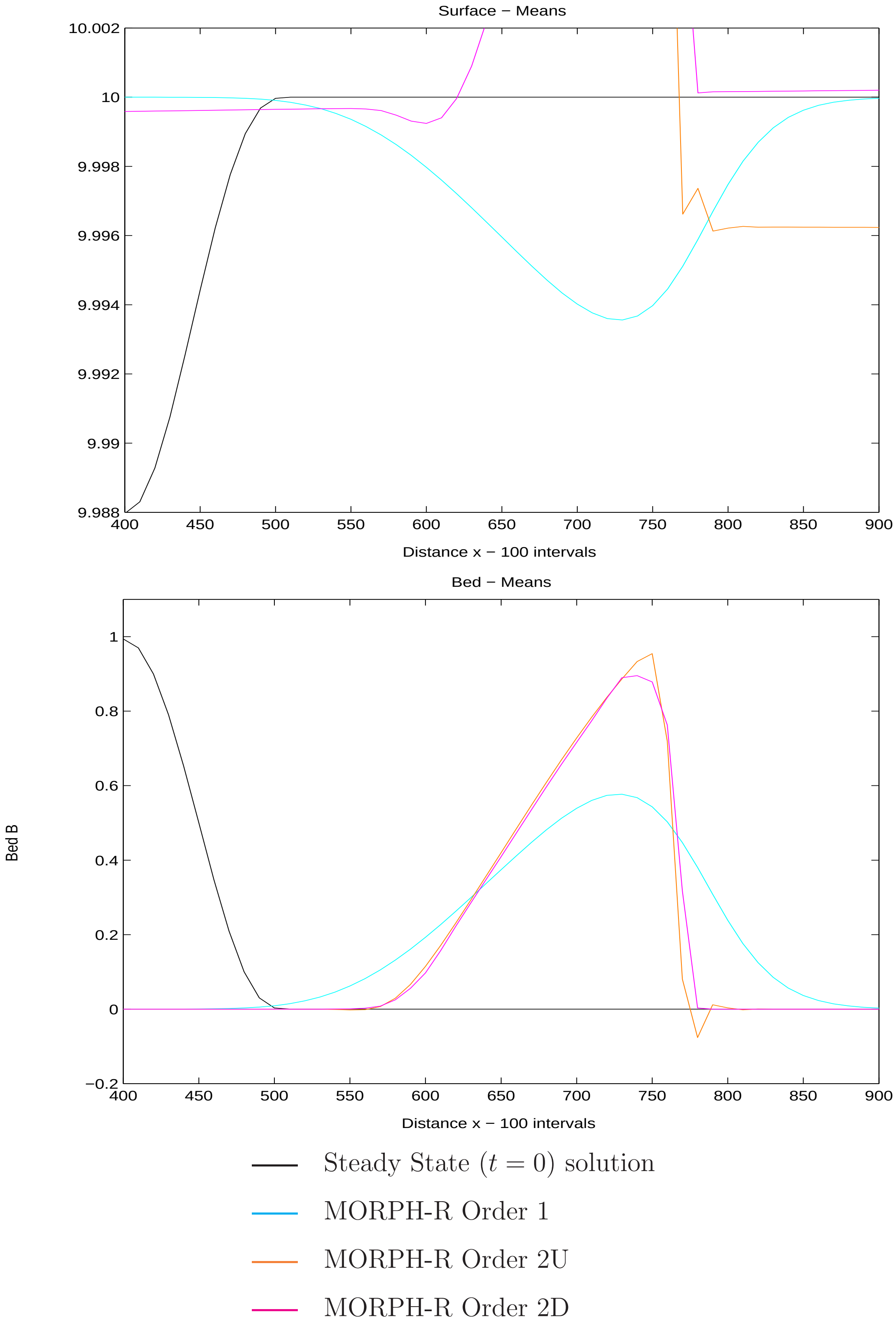


Figure 6.13: Test Case B Results, FD Source Discretisation - MORPH-R

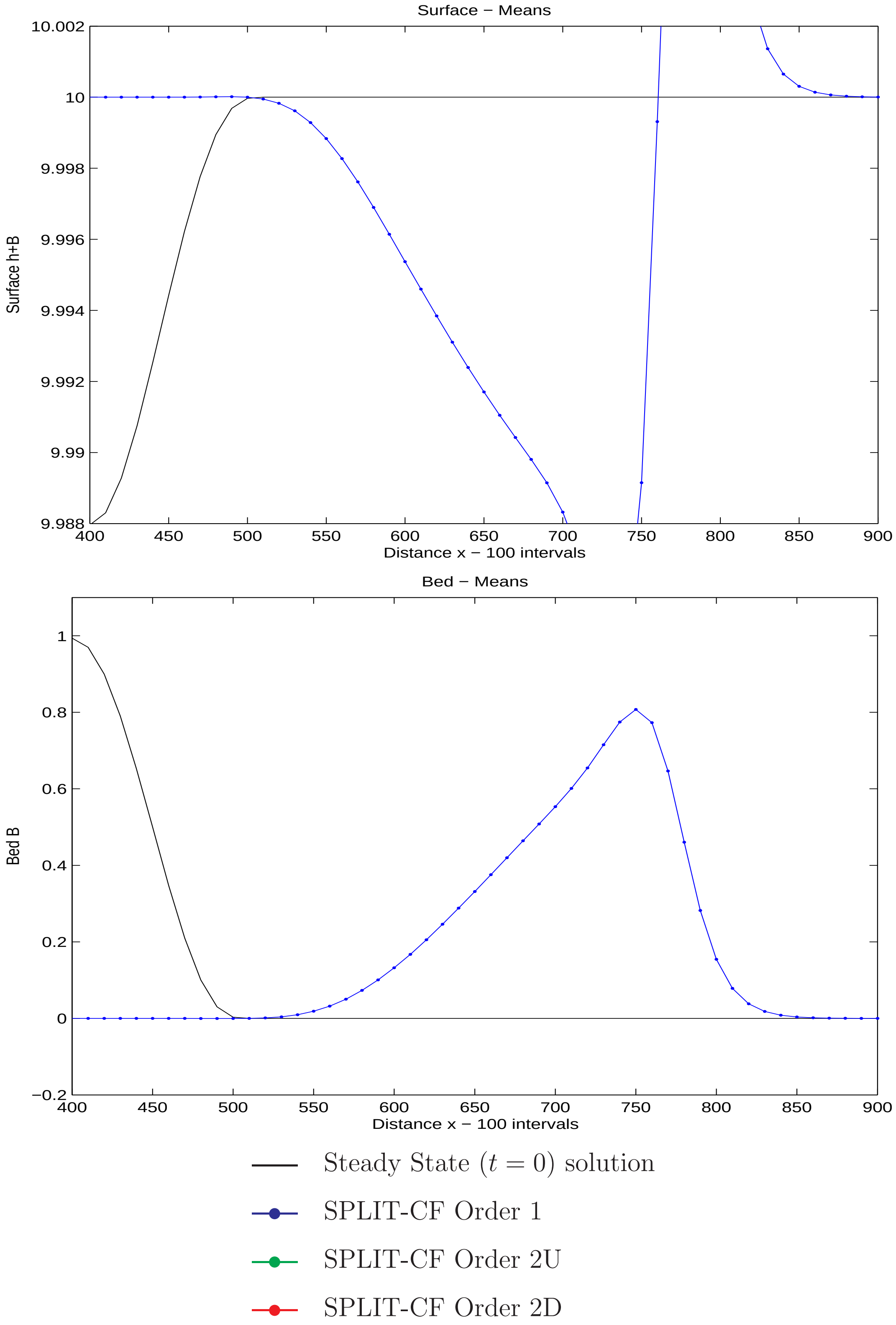


Figure 6.14: Test Case B Results, FD Source Discretisation - SPLIT-CF

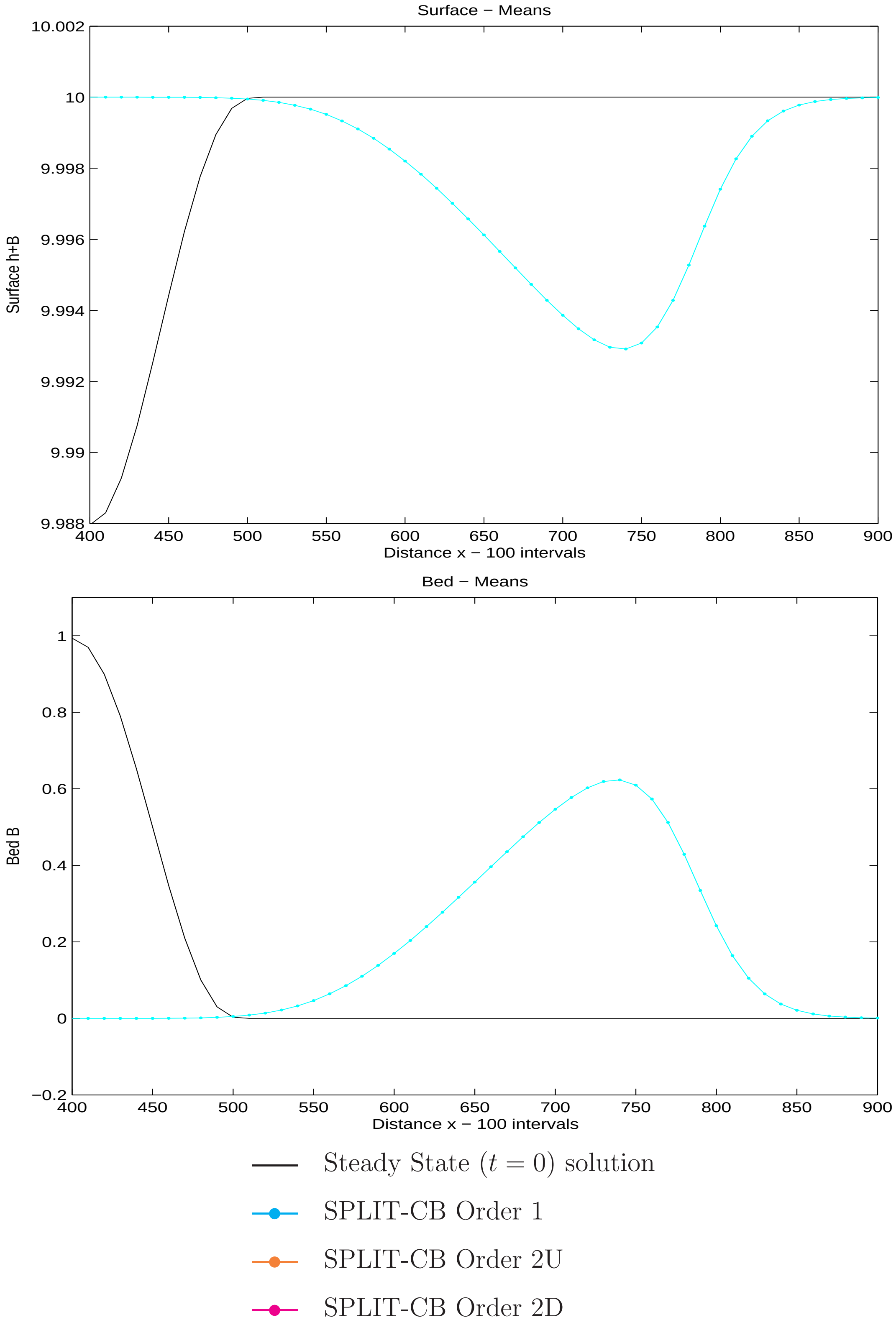


Figure 6.15: Test Case B Results, FD Source Discretisation - SPLIT-CB

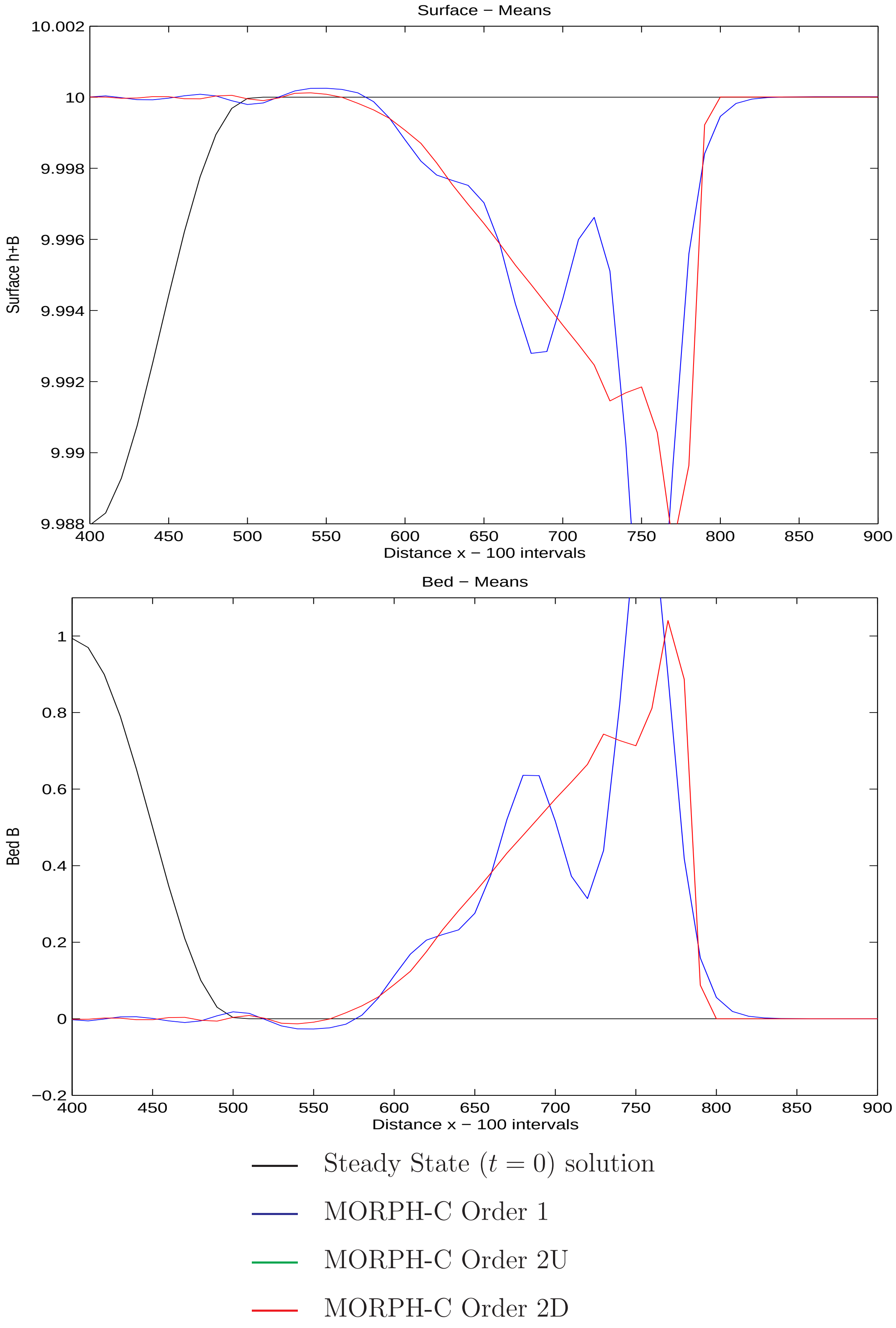


Figure 6.16: Test Case B Results, Proposed Source Discretisation - MORPH-C

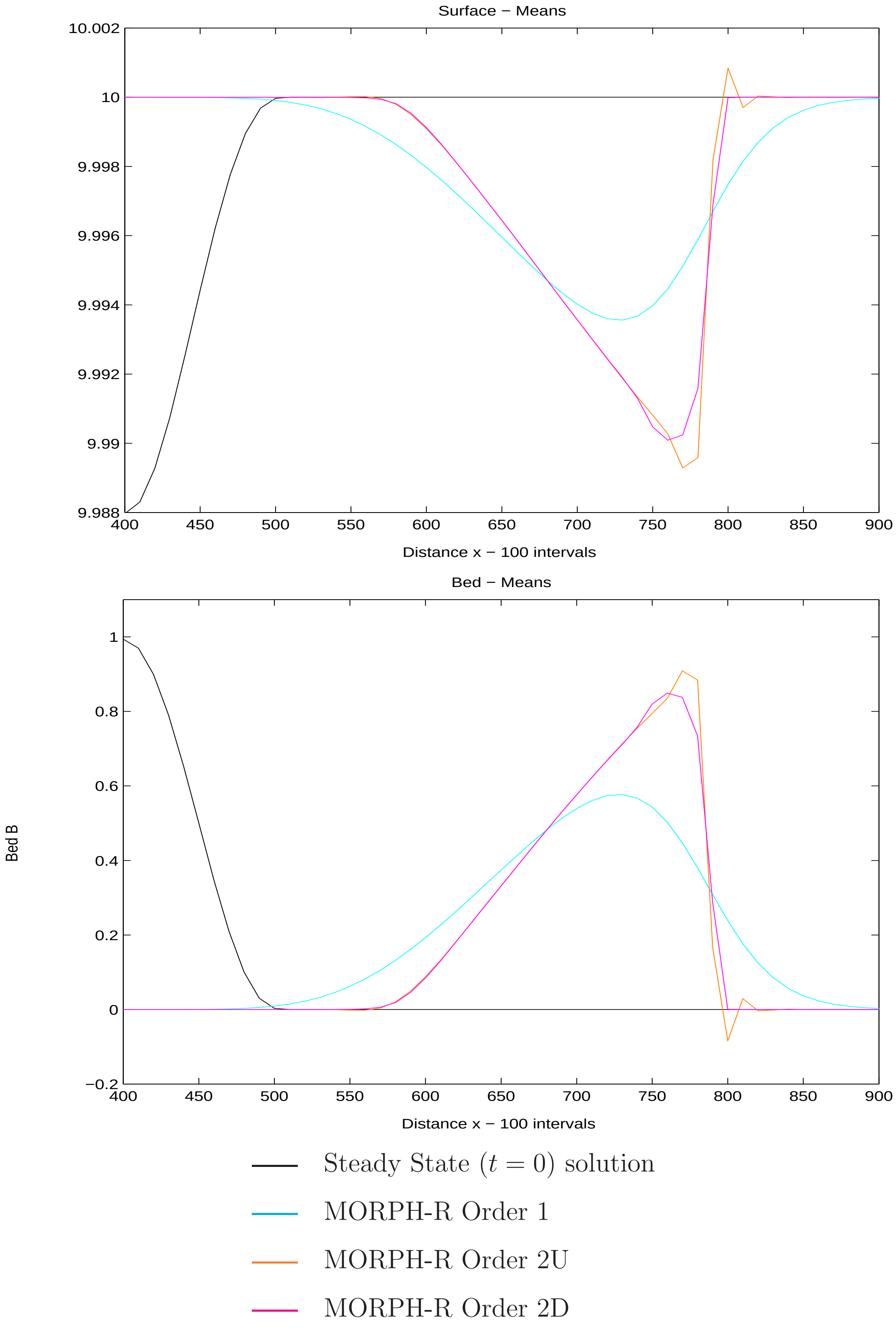


Figure 6.17: Test Case B Results, Proposed Source Discretisation - MORPH-R

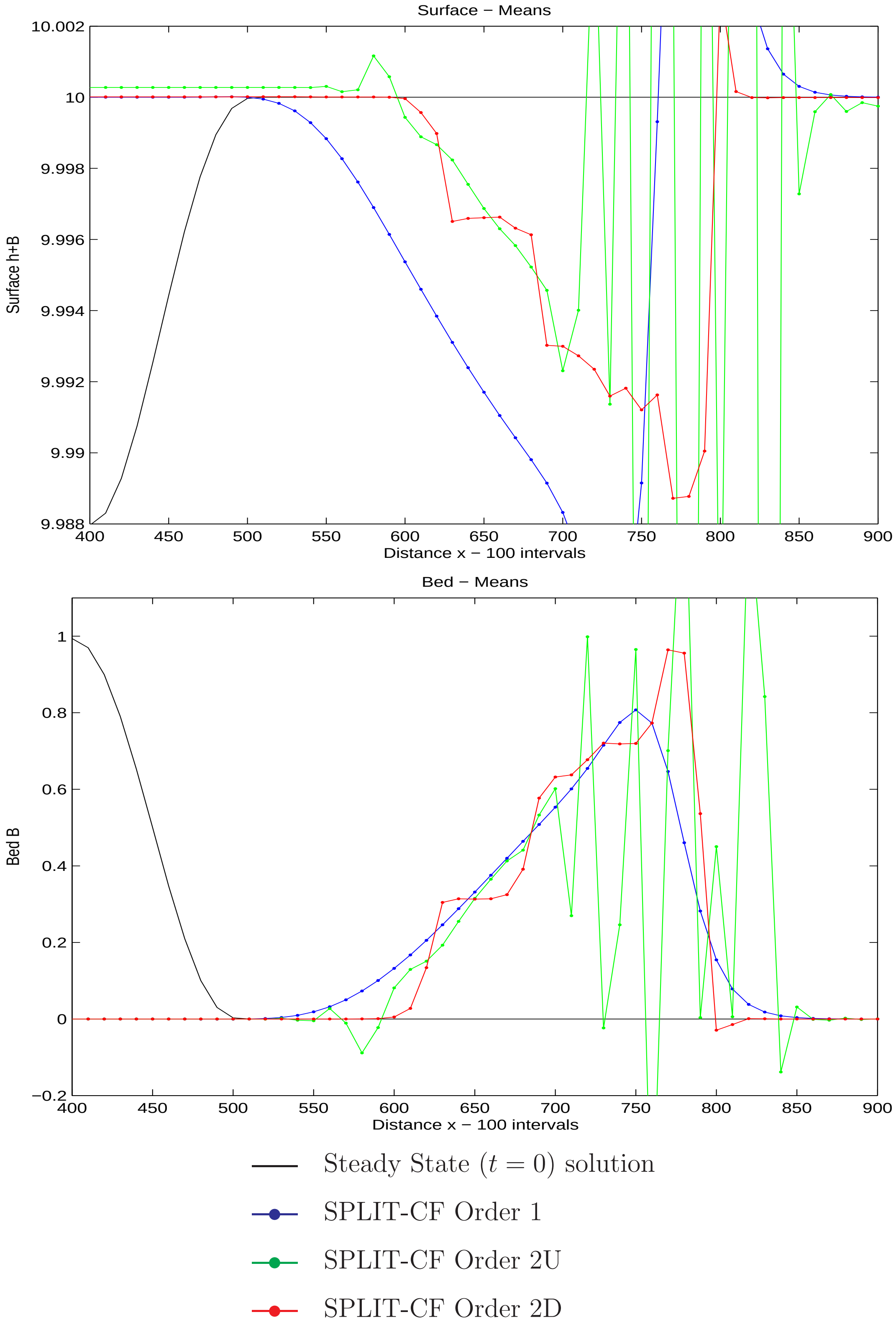


Figure 6.18: Test Case B Results, Proposed Source Discretisation - SPLIT-CF

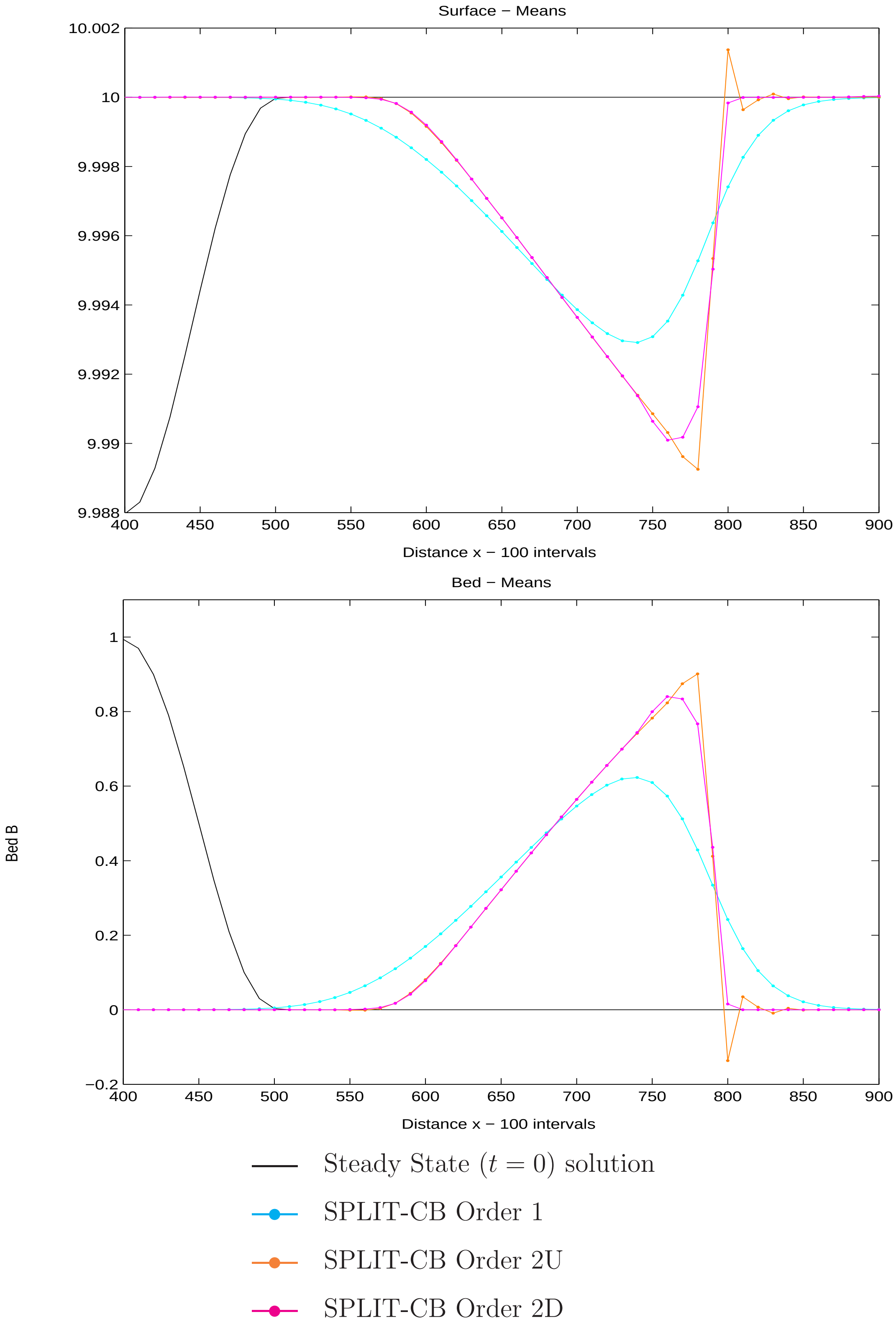


Figure 6.19: Test Case B Results, Proposed Source Discretisation - SPLIT-CB

pears to improve the results and eliminate the oscillations caused by the uninvertible Jacobian.

An important effect, when considering splitting the equations, is the extrapolation used to transfer the solution between time steps. If the water is not given the opportunity to react to the bed movement then it will always cause water flow that is inaccurate. This has been demonstrated with formulation SPLIT-CB performing better, generating smoother results, than formulation SPLIT-CF.

Figure 6.20 shows the results for the second order limited methods for formulation MORPH-R and SPLIT-CB in comparison with the best achievable results from the work by Hudson [34] and the near exact solution, also from Hudson. We can see that the RKDG methods produce results that display a much sharper shock profile, demonstrating the ability of the RKDG scheme to retain shocks within one or two cells [15]. The difference between splitting the equations and solving the complete system appears to be minimal, suggesting that the equations act like they are split. All methods in Figure 6.20 satisfy the C-property.

6.5 More Tests

Having identified a C-property satisfying method we wish to further explore the success of the method. We will introduce an additional test to demonstrate the capabilities of the method.

6.5.1 Test Case C

This test case was introduced by LeVeque [46] as a method of demonstrating the achievements of a numerical method under stiff conditions. The test involves a very small square wave profile passing over a large smooth bump in the bed. This test has been used to demonstrate many methods, for example see [51] or [33].

$$u(x,0) = 0$$

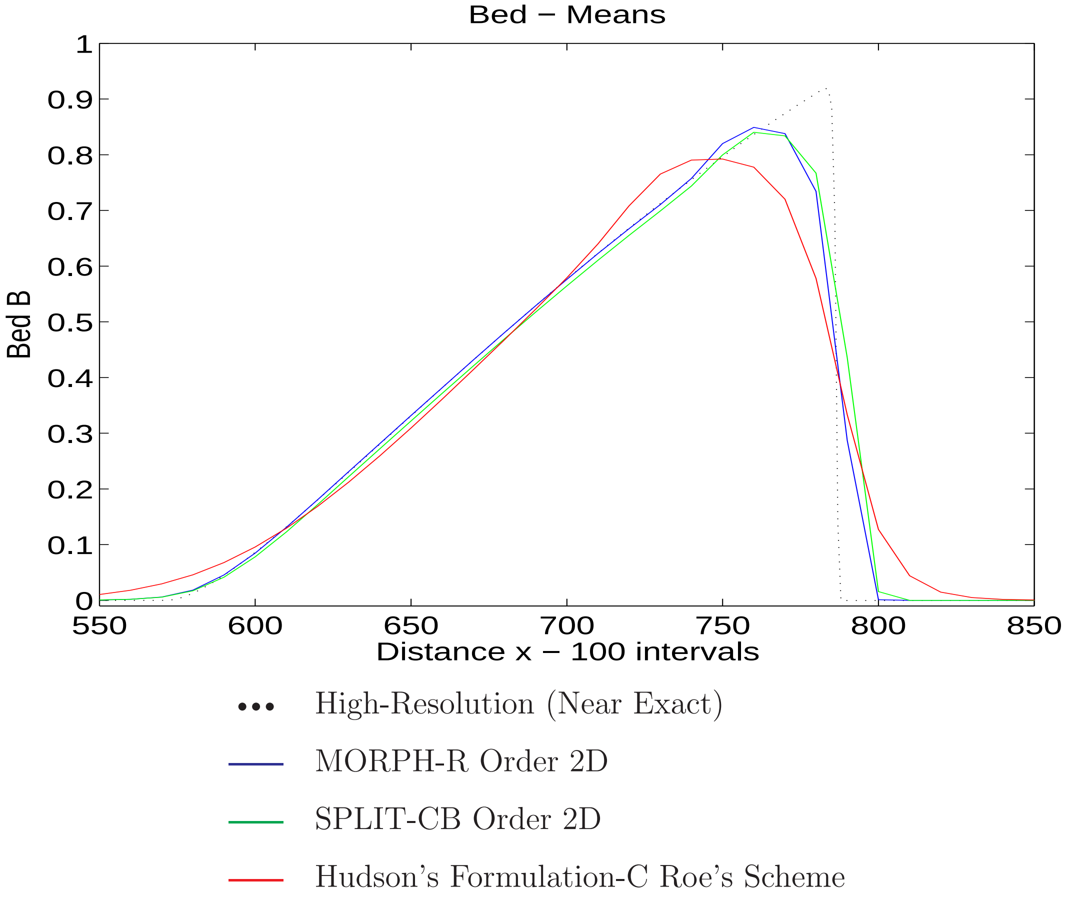


Figure 6.20: Test Case B Results, Comparison of Schemes

$$
\begin{aligned}
h(x,0) &= \begin{cases} 1 - B(x,0) + 0.2, & 0.1 \leq x \leq 0.2, \\ 1 - B(x,0), & \text{otherwise}, \end{cases} \\
B(x,t) &= \begin{cases} 0.25\left(1 + \cos\left(10\pi(x-0.5)\right)\right), & 0.4 \leq x \leq 0.6, \\ 0, & \text{otherwise}. \end{cases}
\end{aligned}
$$

The test will be run on the domain $x = [0, 1]$ and the bed transport coefficient $A = 0$ to maintain a stationary bed. LeVeque defines a unitary gravitational constant, $g = 1$, for this test case so we shall do likewise.

This test is particularly difficult for schemes to model accurately as the the large bed bump creates a large source term. Since the water flow is small, this makes the source term stiff and an error in approximating the source term accurately will generate large changes in the solution. Additionally the square profile of the water bump creates discontinuities in the water profile immediately. The solution of this problem cannot be determined analytically however we can infer qualitative properties of the solution.

It is clear that the bump in the water will split into two outgoing waves of square profile and half the height. These should initially travel at approximately $\sqrt{1.2}$. The left going wave should immediately leave the domain and the right going wave should travel over the bump profile in the bed. As the wave passes over the bump in the bed it should generate a long wave that travels left. In addition, we should see the front of the wave remain a shock but the rear should smooth out into a rarefaction fan.

6.6 Results

The results for Test Case C are shown in Figure 6.21, Figure 6.22 and Figure 6.23.

For the FE source term discretisation, Figure 6.21, it is clear that the first order methods are showing their lack of C-property satisfaction. The Order 2U and Order 2D methods appear give good results and this is due to the bed being, conveniently, continuously represented. The lack of C-property satisfaction means that, not only do we get the bed profile in the surface, we do not represent the left moving long wave at all and this causes the right moving wave to move faster than it should.

For the FD source term discretisation, Figure 6.22, the second order results are demonstrating the lack of C-property satisfaction. This is because the first order methods, conveniently, reduce the representation to piecewise constant. Again, this lack of C-property satisfaction means that, not only do we get the bed profile in the surface, we do not represent the left moving long wave at all and this causes the right moving wave to move faster than it should.

For the proposed source term discretisation, Figure 6.23, we see that none of the methods or formulations have the difficulties of the other source term discretisations. Not only do we not have the bed profile represented in the surface but we also have the left moving long wave and a correctly positioned right moving wave. The Order 1 methods demonstrate the classical diffusion whilst the Order 2U methods demonstrate the classic dispersion and the Order2D methods demonstrate the effectiveness of the limiter.

It is noted that there appears to be a spurious jump for MORPH-R Order 2D in the solution directly above the maximum point of the bed and this is thought to be an effect of the characteristic limiting. As we are limiting $h + B$ and not h the characteristic decomposition is not technically correct for MORPH-R.

To calculate the accuracy, we test the scheme for all formulations at grid resolutions of 100 to 1600. We compare this to a mesh resolution of 16000 and calculate errors using the 1-norm of the mean in cells that correspond to those at a grid resolution of 100 cells. The errors and orders of accuracy are given in Table 6.1 to Table 6.5.

It is clear that the methods are all converging to a solution which appears to be correct, however the difficulty of the test case makes the numbers somewhat poor reflectors of the methods' success. This is demonstrated through the fact that comparing the order of accuracies between Order 1, Order 2U and Order2D do not give any coherent pattern yet comparison of different formulations give very similar results.

SWE-C	Order 1		Order 2U		Order 2D	
Grid	Error	Order	Error	Order	Error	Order
100	0.10788		0.05502		0.05502	
200	0.07562	0.51260	0.03790	0.53772	0.03495	0.65465
400	0.05143	0.55622	0.01084	1.80614	0.01588	1.13811
800	0.03133	0.71508	0.00463	1.22626	0.00434	1.87150
1600	0.01492	1.07017	0.00247	0.90538	0.00168	1.36629

Table 6.1: Comparison of Error and Order for Formulation SWE-C

MORPH-C	Order 1		Order 2U		Order 2D	
Grid	Error	Order	Error	Order	Error	Order
100	0.10788		0.05502		0.05502	
200	0.07562	0.51260	0.03790	0.53772	0.03495	0.65465
400	0.05143	0.55622	0.01084	1.80614	0.01588	1.13810
800	0.03133	0.71508	0.00463	1.22626	0.00434	1.87150
1600	0.01492	1.07017	0.00247	0.90538	0.00168	1.36629

Table 6.2: Comparison of Error and Order for Formulation MORPH-C

MORPH-R	Order 1		Order 2U		Order 2D	
Grid	Error	Order	Error	Order	Error	Order
100	0.11220		0.05501		0.05298	
200	0.07859	0.51357	0.03782	0.54040	0.03587	0.56270
400	0.05301	0.56791	0.01080	1.80881	0.01624	1.14279
800	0.03209	0.72418	0.00464	1.21952	0.00438	1.89128
1600	0.01537	1.06212	0.00246	0.91206	0.00170	1.36451

Table 6.3: Comparison of Error and Order for Formulation MORPH-R

SPLIT-CB	Order 1		Order 2U		Order 2D	
Grid	Error	Order	Error	Order	Error	Order
100	0.10393		0.04468		0.04601	
200	0.07388	0.49245	0.03604	0.31014	0.03254	0.49967
400	0.04863	0.60324	0.01053	1.77566	0.01417	1.19901
800	0.02990	0.70189	0.00461	1.19033	0.00425	1.73618
1600	0.01414	1.08066	0.00247	0.90042	0.00166	1.35602

Table 6.4: Comparison of Error and Order for Formulation SPLIT-CB

SPLIT-CF	Order 1		Order 2U		Order 2D	
Grid	Error	Order	Error	Order	Error	Order
100	0.10393		0.04468		0.04601	
200	0.07388	0.49245	0.03604	0.31014	0.03254	0.49967
400	0.04863	0.60324	0.01053	1.77566	0.01417	1.19901
800	0.02990	0.70189	0.00461	1.19033	0.00425	1.73618
1600	0.01414	1.08066	0.00247	0.90042	0.00166	1.35602

Table 6.5: Comparison of Error and Order for Formulation SPLIT-CF

We can see that both formulation SWE-C and MORPH-C agree to machine precision for error and accuracy reflecting the fact that, with the bed constant, A at zero the formulations are actually identical. Additionally, both SPLIT-CF and SPLIT-CB also agree however they do not agree with SWE-C and MORPH-C. It is believed that this is caused by difference when boundary conditions and limiters are applied in the algorithm.

The order of accuracies across the Order 1 methods generally show an accuracy order of between 0.5 and 1. The order 2U and order2D methods show an accuracy order of between 0.9 and 1.8 suggesting greater than order 1 convergence.

6.7 Summary

In this chapter we have proposed a source term discretisation that combines a finite element and finite difference approach to balance the flux discretisation. This source term discretisation has been proven to be C-property satisfying and numerical results have demonstrated that this is so in practice.

We have also also considered the effect of two-speed time stepping and proposed four interpretations that can be used for extrapolating the solution between time steps. We have demonstrated results for two of these extrapolation methods and shown that backwards extrapolation is much more accurate than the usual forwards extrapolation.

We have seen that two formulations, MORPH-R and SPLIT-CB, produce good results for morphodynamics when the proposed source term discretisation is used with any order of accuracy. We have, thus, given a method for modelling morphodynamics in a split and a combined approach.

In the next chapter we will extend the RKDG method to 2D and give details of how the methods and ideas that have been used in 1D apply in 2D.

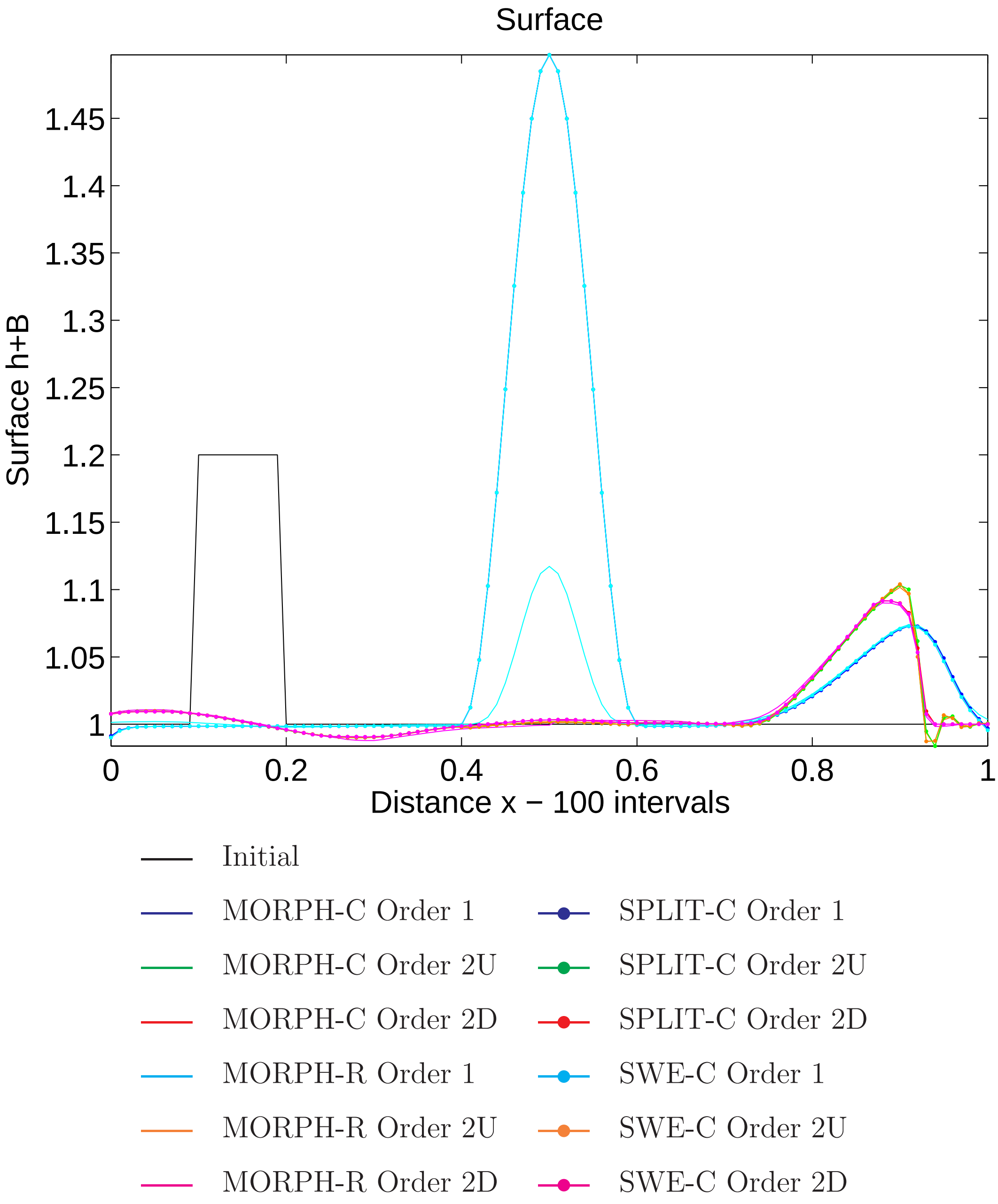


Figure 6.21: Test Case C Results - FE Discretisation

SWE-C, SPLIT-C and MORPH-C Order 1 are identical and are represented by the cyan dotted line. All Order 2U results are similar and are close to the magenta dotted line. All Order 2D results are similar and are close to the orange dotted line.

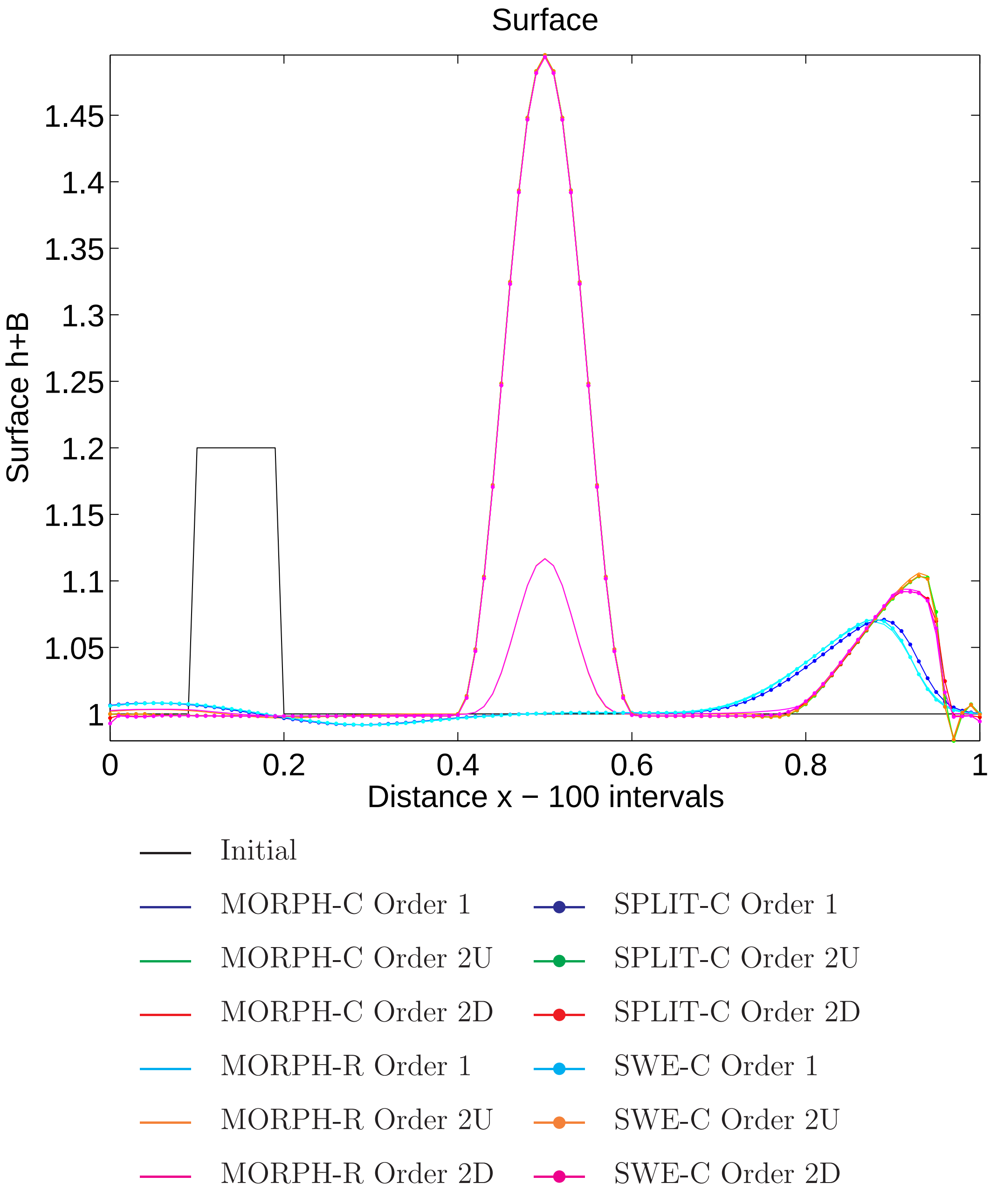


Figure 6.22: Test Case C Results - FD Discretisation

All Order 1 results are similar and are close to the cyan dotted line. All Order 2U results are similar and are close to the orange dotted line. All Order 2D results are similar and are close to the magenta dotted line. In the centre all Order 2U and Order 2D results are the same and are represented by the magenat dotted line.

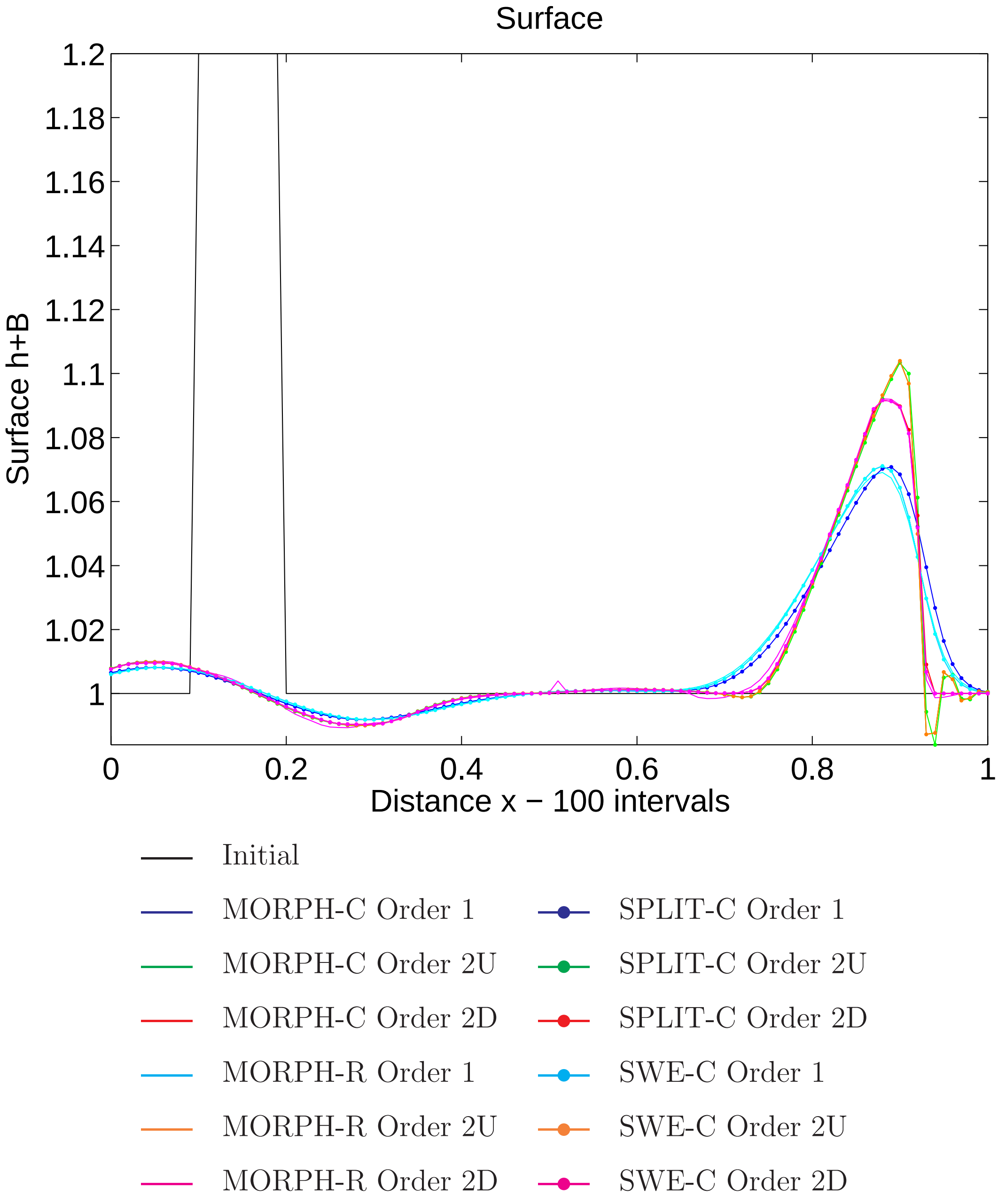


Figure 6.23: Test Case C Results - Proposed Discretisation

Chapter 7

Extension to 2D

In the series of papers by Cockburn *et al.* we saw the RKDG method developed from a 1D scalar equation [16] to systems [15], multidimensional scalars [13] and finally multidimensional systems [18]. The 2D shallow water and morphodynamical equations are multidimensional systems with source terms. We need to take the RKDG method at its most complex and extend it to enable the modelling of the equations of interest.

The SWE and morphodynamical equations have 2D equivalents which describe the flow in a region so we can easily model a region such as a river or estuary with the 2D equations. It is also fortunate that, unlike finite differences, finite elements naturally extends to multiple dimensions and are easily applied to an unstructured mesh. Many of the ideas that we saw in 1D automatically apply directly to 2D. In this chapter we shall give the extension of the 1D formulations in 2D and outline the elements needed to create the 2D RKDG method. We will also apply our knowledge gained from 1D to create the extensions to the RKDG method and show that the method, with the extensions, can be successfully applied to the SWE and morphodynamical equations.

7.1 Formulations

We can extend the formulations given in 1D to 2D by naturally adding an additional spatial dimension, y. In this case the velocity attains a direction and has two components. We label u the velocity component in the x direction and v the velocity component in the y direction. We also label $Qx = hu$ and $Qy = hv$ the discharges in the x and y directions respectively.

Since we effectively now have an additional parameter to solve for we need an additional equation and this is given by considering the conservation of momentum in the y direction. In the following sections we outline the extensions of the formulations we used in 1D. In all cases the 1D formulations can be recovered by enforcing $v \equiv 0$.

We will express each formulation in the form,

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F} + \frac{\partial}{\partial y}\mathbf{G} = \mathbf{R},$$

or,

$$\frac{\partial}{\partial t}\mathbf{U} + \underline{\nabla}.[\mathbf{F}, \mathbf{G}]^T = \mathbf{R}. \tag{7.1}$$

7.1.1 Formulation 2DSWE-C

The extension of formulation SWE-C to formulation 2DSWE-C is given by,

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} hu \\ {}^1\!/\!_2 gh^2 + hu^2 \\ huv \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} hv \\ huv \\ {}^1\!/\!_2 gh^2 + hv^2 \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0 \\ -gh\frac{\partial B}{\partial x} \\ -gh\frac{\partial B}{\partial y} \end{bmatrix}.$$

This system has Jacobians,

$$\mathrm{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 \\ gh - u^2 & 2u & 0 \\ -uv & v & u \end{bmatrix},$$

and,

$$\mathrm{B} = \frac{\partial \mathbf{G}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ gh - v^2 & 0 & 2v \end{bmatrix},$$

whose inversions are

$$\mathrm{A}^{-1} = \frac{1}{u(gh - u^2)} \begin{bmatrix} -2u^2 & u & 0 \\ u(gh - u^2) & 0 & 0 \\ -v(gh + u^2) & uv & gh - u^2 \end{bmatrix},$$

and

$$\mathrm{B}^{-1} = \frac{1}{v(gh - v^2)} \begin{bmatrix} -2v^2 & 0 & v \\ -u(gh + v^2) & gh - v^2 & uv \\ v(gh - v^2) & 0 & 0 \end{bmatrix}.$$

The matrix A has an eigendecomposition of,

$$\Lambda^{(A)} = \begin{bmatrix} u - c & 0 & 0 \\ 0 & u & 0 \\ 0 & 0 & u + c \end{bmatrix}, \qquad \mathrm{X}^{(A)} = \begin{bmatrix} 1 & 0 & 1 \\ u - c & 0 & u + c \\ v & c & v \end{bmatrix},$$

and the matrix B has an eigendecomposition of,

$$\Lambda^{(A)} = \begin{bmatrix} v - c & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & v + c \end{bmatrix}, \qquad \mathrm{X}^{(A)} = \begin{bmatrix} 1 & 0 & 1 \\ u & c & u \\ v - c & 0 & v + c \end{bmatrix}.$$

We will also need the eigendecomposition of the matrix formed from a combination of the Jacobian matrices. For formulation 2DSWE-C this is given by,

$$\begin{bmatrix} \mathrm{A} \\ \mathrm{B} \end{bmatrix} .\mathbf{n} = \begin{bmatrix} 0 & 1.n_x & 1.n_y \\ c^2.n_x - u^2.n_x - uv.n_y & 2u.n_x + v.n_y & u.n_y \\ -uv.n_x + c^2.n_y - v^2.n_y & v.n_x & u.n_x + 2v.n_y \end{bmatrix}.$$

where $\mathbf{n}$ is a unit vector, in our case it will be a normal to an edge. The eigendecomposition of this matrix is given by [22] as

$$\Lambda = \begin{bmatrix} u.n_x + v.n_y - c & 0 & 0 \\ 0 & u.n_x + v.n_y & 0 \\ 0 & 0 & u.n_x + v.n_y + c \end{bmatrix},$$

$$\mathrm{X} = \begin{bmatrix} 1 & 0 & 1 \\ u - c.n_x & n_y & u + c.n_x \\ v - c.n_y & -n_x & v + c.n_y \end{bmatrix}.$$

Later, we shall need the value of $|[\mathrm{A}, \mathrm{B}]^T.\mathbf{n}|$ under the assumption that $u = v = 0$, which we determine here for simplicity. We begin with,

$$|\Lambda| = \begin{bmatrix} c & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & c \end{bmatrix},$$

and,

$$\mathrm{X} = \begin{bmatrix} 1 & 0 & 1 \\ -c.n_x & n_y & c.n_x \\ -c.n_y & -n_x & c.n_y \end{bmatrix}.$$

This means that,

$$\mathrm{X}^{-1} = \frac{1}{2c} \begin{bmatrix} c & -n_x & -n_y \\ 0 & 2c.n_y & -2c.n_x \\ c & n_x & n_y \end{bmatrix},$$

using the fact that ${n_x}^2 + {n_y}^2 = 1$. From these we can see that,

$$\mathrm{X}|\Lambda| = \begin{bmatrix} c & 0 & c \\ -c^2.n_x & 0 & c^2.n_x \\ -c^2.n_y & 0 & c^2.n_y \end{bmatrix}.$$

and,

$$\mathrm{X}|\Lambda|\mathrm{X}^{-1} = \frac{1}{2c} \begin{bmatrix} 2c^2.({n_x}^2 + {n_y}^2) & 0 & 0 \\ 0 & 2c^2.{n_x}^2 & 2c^2.n_x.n_y \\ 0 & 2c^2.n_x.n_y & 2c^2.{n_y}^2 \end{bmatrix}.$$

This means that,

$$\left| \begin{bmatrix} \mathrm{A} \\ \mathrm{B} \end{bmatrix} .\mathbf{n} \right| = \mathrm{X}|\Lambda|\mathrm{X}^{-1} = \begin{bmatrix} c & 0 & 0 \\ 0 & c.{n_x}^2 & c.n_x.n_y \\ 0 & c.n_x.n_y & c.{n_y}^2 \end{bmatrix}. \qquad (7.2)$$

7.1.2 The Morphodynamical Equation

In 1D the morphodynamical equation was difficult to define. We used an approximation to the equation that was very simple to express. In 2D the process becomes more difficult. The question of what exactly the morphodynamical equation should be is an ongoing discussion, for examples see [34]. Many suggestions have been put forward and we choose to follow Hudson [34] in using,

$$\frac{\partial}{\partial t}B + A\frac{\partial}{\partial x}q_1(h,u,v) + A\frac{\partial}{\partial y}q_2(h,u,v) = 0,$$

where,

$$\begin{aligned} q_1(h,u,v) &= \xi u(u^2+v^2), \\ q_2(h,u,v) &= \xi v(u^2+v^2). \end{aligned}$$

7.1.3 Formulation 2DMORPH-C

It was clear from the results shown in Section 6.4.3 that formulation MORPH-C cannot produce good results for morphodynamics. We do not intend on using the 2D equivalent of this formulation, we merely give it here for completeness. No results will be produced using this formulation.

The extension of formulation MORPH-C to formulation 2DMORPH-C is given by,

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \\ B \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} hu \\ \tfrac{1}{2}gh^2 + hu^2 \\ huv \\ A\xi u(u^2+v^2) \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} hv \\ huv \\ \tfrac{1}{2}gh^2 + hv^2 \\ A\xi v(u^2+v^2) \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0 \\ -gh\frac{\partial B}{\partial x} \\ -gh\frac{\partial B}{\partial y} \\ 0 \end{bmatrix}.$$

This system has Jacobians of

$$\mathrm{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ gh - u^2 & 2u & 0 & 0 \\ -uv & v & u & 0 \\ -3A\xi u(u^2+v^2)/h & A\xi(3u^2+v^2)/h & 2A\xi uv/h & 0 \end{bmatrix},$$

and,

$$\mathrm{B} = \frac{\partial \mathbf{G}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ gh - v^2 & 0 & 2v & 0 \\ -3A\xi v(u^2+v^2)/h & 2A\xi uv/h & A\xi(u^2+3v^2)/h & 0 \end{bmatrix}.$$

The matrix A has an eigendecomposition of,

$$\Lambda^{(A)} = \begin{bmatrix} u-c & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & u+c \end{bmatrix}, \quad \mathrm{X}^{(A)} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ u-c & 0 & 0 & u+c \\ v & 0 & 1 & v \\ \frac{A\xi c(3u^2+v^2)}{h(c-u)} & c & \frac{2A\xi v}{h} & \frac{A\xi c(3u^2+v^2)}{h(c+u)} \end{bmatrix},$$

while the matrix B has an eigendecomposition of,

$$\Lambda^{(B)} = \begin{bmatrix} v-c & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 0 & v+c \end{bmatrix}, \quad \mathrm{X}^{(B)} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ u & 0 & 1 & u \\ v-c & 0 & 0 & v+c \\ \frac{A\xi c(u^2+3v^2)}{h(c-v)} & c & \frac{2A\xi u}{h} & \frac{A\xi c(u^2+3v^2)}{h(c+v)} \end{bmatrix}.$$

We will also provide the eigendecomposition of the matrix formed from a combination of the Jacobian matrices. For formulation 2DMORPH-C this is given by,

$$\begin{bmatrix} \mathrm{A} \\ \mathrm{B} \end{bmatrix} .\mathbf{n} = \begin{bmatrix} 0 & n_x & n_y & 0 \\ c^2.n_x - u^2.n_x - uv.n_y & 2u.n_x + v.n_y & u.n_y & 0 \\ -uv.n_x + c^2.n_y - v^2.n_y & v.n_x & u.n_x + 2v.n_y & 0 \\ d_1 & d_2 & d_3 & 0 \end{bmatrix},$$

where,

$$d_1 = -3A\xi(u.n_x + v.n_y)(u^2+v^2)/h,$$

$$d_2 = A\xi(3u^2.n_x + v^2.n_x + 2uv.n_y)/h,$$

$$d_3 = A\xi(2uv.n_x + u^2.n_y + 3v^2.n_y)/h,$$

and **n** is a unit vector, in our case it will be a normal to an edge. The eigendecomposition of this matrix is given by,

$$\Lambda = \begin{bmatrix} u.n_x + v.n_y - c & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & u.n_x + v.n_y & 0 \\ 0 & 0 & 0 & u.n_x + v.n_y + c \end{bmatrix},$$

$$\mathrm{X} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ u - c.n_x & 0 & n_y & u + c.n_x \\ v - c.n_y & 0 & -n_x & v + c.n_y \\ A\xi c\frac{u^2+v^2+2(n_xu+n_yv)^2}{h(c-n_xu-n_yv)} & c & 2A\xi\frac{(u.n_y - v.n_x)}{h} & A\xi c\frac{u^2+v^2+2(n_xu+n_yv)^2}{h(c+n_xu+n_yv)} \end{bmatrix}.$$

7.1.4 Formulation 2DMORPH-R

The extension of formulation MORPH-R to formulation 2DMORPH-R is given by,

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \\ B \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} hu \\ {}^1\!/_2 gh(h+2B) + hu^2 \\ huv \\ A\xi u(u^2+v^2) \end{bmatrix},$$

$$\mathbf{G} = \begin{bmatrix} hv \\ huv \\ {}^1\!/_2 gh(h+2B) + hv^2 \\ A\xi v(u^2+v^2) \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} 0 \\ gB\frac{\partial h}{\partial x} \\ gB\frac{\partial h}{\partial y} \\ 0 \end{bmatrix}.$$

This system has a Jacobian of,

$$\mathrm{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ g(h+B)-u^2 & 2u & 0 & gh \\ -uv & v & u & 0 \\ -3A\xi u(u^2+v^2)/h & A\xi(3u^2+v^2)/h & 2A\xi uv/h & 0 \end{bmatrix}.$$

Eigenvalues are $\lambda^{(3A)} = u$ and three others, $\lambda^{(1A)} \approx u-c$, $\lambda^{(2A)} \approx 0$ and $\lambda^{(4A)} \approx u+c$, given by the roots of the polynomial,

$$\lambda^3 + (-2u)\lambda^2 + (u^2 - g(h+B) - A\xi g(3u^2+v^2))\lambda + (A\xi gu(3u^2+v^2)).$$

We use the algorithm given in Section 2.3.5 to determine these eigenvalues. The matrix A has an eigendecomposition of,

$$\Lambda^{(A)} = \begin{bmatrix} \lambda^{(1A)} & 0 & 0 & 0 \\ 0 & \lambda^{(2A)} & 0 & 0 \\ 0 & 0 & u & 0 \\ 0 & 0 & 0 & \lambda^{(4A)} \end{bmatrix},$$

$$\mathrm{X}^{(A)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \lambda^{(1A)} & \lambda^{(2A)} & u & \lambda^{(4A)} \\ v & v & v - \frac{h+B}{2A\xi v} & v \\ z^{(A)}(\lambda^{(1A)}) & z^{(A)}(\lambda^{(2A)}) & -\frac{h+B}{h} & z^{(A)}(\lambda^{(4A)}) \end{bmatrix},$$

where,

$$z^{(A)}(\lambda) = \frac{u^2 - g(h+B) + (\lambda - 2u)\lambda}{gh}.$$

If v is near zero then the term $v - (h+B)/(2A\xi v)$ becomes large. At small values of v we replace the third column with the limiting value $[0,0,-1,0]^T$ to avoid extreme variations of orders of magnitude of the terms in the matrix.

Likewise,

$$\mathrm{B} = \frac{\partial \mathbf{G}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ g(h+B)-v^2 & 0 & 2v & gh \\ -3A\xi v(u^2+v^2)/h & 2A\xi uv/h & A\xi(u^2+3v^2)/h & 0 \end{bmatrix},$$

whose eigenvalues are $\lambda^{(3B)} = v$ and three others, $\lambda^{(1B)} \approx v - c$, $\lambda^{(2B)} \approx 0$ and $\lambda^{(4B)} \approx v + c$, given by the roots of the polynomial,

$$\lambda^3 + (-2v)\lambda^2 + (v^2 - g(h+B) - A\xi g(u^2+3v^2))\lambda + (A\xi gv(u^2+3v^2)).$$

We use the algorithm given in Section 2.3.5 to determine these eigenvalues. The matrix B has an eigendecomposition of,

$$\Lambda^{(B)} = \begin{bmatrix} \lambda^{(1B)} & 0 & 0 & 0 \\ 0 & \lambda^{(2B)} & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 0 & \lambda^{(4B)} \end{bmatrix},$$

$$\mathrm{X}^{(B)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ u & u & u - \frac{h+B}{2A\xi u} & u \\ \lambda^{(1B)} & \lambda^{(2B)} & v & \lambda^{(4B)} \\ z^{(B)}(\lambda^{(1B)}) & z^{(B)}(\lambda^{(2B)}) & -\frac{h+B}{h} & z^{(A)}(\lambda^{(4B)}) \end{bmatrix},$$

where,

$$z^{(B)}(\lambda) = \frac{v^2 - g(h+B) + (\lambda - 2v)\lambda}{gh}.$$

If u is near zero then the term $u - (h+B)/(2A\xi u)$ becomes large. At small values of u we replace the third column with the limiting value $[0, -1, 0, 0]^T$ to avoid extreme variations of orders of magnitude of the terms in the matrix

We will also need the eigendecomposition of the matrix formed from a combination of the Jacobian matrices. For formulation 2DMORPH-R this is given by,

$$\begin{bmatrix} \mathrm{A} \\ \mathrm{B} \end{bmatrix} .\mathbf{n} = \begin{bmatrix} 0 & n_x & n_y & 0 \\ (gh + gB - u^2).n_x - uv.n_y & 2u.n_x + v.n_y & u.n_y & gh.n_x \\ -uv.n_x + (gh + gB - v^2).n_y & v.n_x & u.n_x + 2v.n_y & gh.n_y \\ d_1 & d_2 & d_3 & 0 \end{bmatrix},$$

where,

$$d_1 = -3A\xi(u.n_x + v.n_y)(u^2 + v^2)/h,$$

$$d_2 = A\xi(3u^2.n_x + v^2.n_x + 2uv.n_y)/h,$$

$$d_3 = A\xi(2uv.n_x + u^2.n_y + 3v^2.n_y)/h,$$

and **n** is a unit vector, in our case it will be a normal to an edge. Unfortunately, unlike the previous formulations, the eigendecomposition of this matrix is not simple to determine but is given by,

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & u.n_x + v.n_y & 0 \\ 0 & 0 & 0 & \lambda_4 \end{bmatrix},$$

$$\mathrm{X} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ a_{11} & a_{21} & u + \frac{(h+B).n_y}{2A\xi(v.n_x - u.n_y)} & a_{41} \\ a_{12} & a_{22} & v - \frac{(h+B).n_x}{2A\xi(v.n_x - u.n_y)} & a_{42} \\ a_{13} & a_{23} & -\frac{h+B}{h} & a_{43} \end{bmatrix}.$$

The undetermined eigenvalues are given by the algorithm in Section 2.3.5 applied to the polynomial,

$$\begin{gathered} \lambda^3 \\ + (-2u.n_x - 2v.n_y)\,\lambda^2 \\ + \Big((u.n_x + v.n_y)^2 - g(h+B) - A\xi g((u^2+v^2) + 2(u.n_x + v.n_y)^2)\Big)\,\lambda \\ + \Big(A\xi g(u.n_x + v.n_y)((u^2+v^2) + 2(u.n_x + v.n_y)^2)\Big) = 0, \end{gathered}$$

and a_{ij} are given by the solution of the reduced system,

$$\begin{bmatrix} 2u.n_x + v.n_y - \lambda_i & u.n_y & gh.n_x \\ v.n_x & u.n_x + 2v.n_y - \lambda_i & gh.n_y \\ d_2 & d_3 & -\lambda_i \end{bmatrix} \begin{bmatrix} a_{i1} \\ a_{i2} \\ a_{i3} \end{bmatrix} = \begin{bmatrix} uv.n_y - (gh + gB - u^2).n_x \\ uv.n_x - (gh + gB - v^2).n_y \\ -d_1 \end{bmatrix}.$$

If $v.n_x - u.n_y$ is near zero then the middle terms in the third column of X become large. At the limit we replace the third column with $[0, n_y, -n_x, 0]^T$.

7.1.5 Formulation 2DSPLIT-C

We can apply the same technique of separating the time scales as we did in 1D. The equations we shall approximate are given by,

$$\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} hu \\ \tfrac{1}{2}gh^2 + hu^2 \\ huv \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} hv \\ huv \\ \tfrac{1}{2}gh^2 + hv^2 \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} 0 \\ -gh\frac{\partial B}{\partial x} \\ -gh\frac{\partial B}{\partial y} \end{bmatrix},$$

along with,

$$\frac{\partial B}{\partial t} + A\xi\frac{\partial}{\partial x}(u(u^2+v^2)) + A\xi\frac{\partial}{\partial y}(v(u^2+v^2)) = 0.$$

The hydrodynamic system has a Jacobian and eigendecomposition identical to formulation 2DSWE-C. We again have a flux function that is not a direct function of B so we follow the same approach as we used in 1D to give,

$$\lambda_x \approx \frac{3A\xi Qx_c({Qx_c}^2 + {Qy_c}^2)}{(D-B)^4}, \qquad \lambda_y \approx \frac{3A\xi Qy_c({Qx_c}^2 + {Qy_c}^2)}{(D-B)^4}.$$

The combination of which is,

$$\begin{bmatrix} \lambda_x \\ \lambda_y \end{bmatrix} . \mathbf{n} = \frac{3A\xi(Qx_c.n_x + Qy_c.n_y)({Qx_c}^2 + {Qy_c}^2)}{(D-B)^4}.$$

7.2 Preliminaries

The extension, from 1D to 2D, of the concepts discussed in Chapter 3 is fairly simple. The concept of conservation still applies in 2D. We should expect the method to conserve mass and momentum in the presence of zero net inflow. The C-property in 2D will be discussed in Section 7.7.1 and the concept of TVD will be discussed in Section 7.4. The principles of CFL limits, stability and adaptive time stepping will be carried forward.

Non-dimensionalisation in 2D is achieved through the same manner as 1D. The non-dimensionalisation parameter L is defined to be the maximum width in either of the x or y directions. Non-dimensionalisation is then achieved in the same manner as before.

The inclusion of a second velocity component, v, and the enlargement of the system means that we need to consider the Roe-averages again. It is fortunate that the extension to 2D is simple. We need to include a Roe-average for v and it is actually analogous to u. This means that the Roe-averages, in 2D, are given by,

$$\begin{aligned}\bar{h} &= {}^1\!/_2(h_{_L} + h_{_R}),\\[1em] \bar{B} &= {}^1\!/_2(B_{_L} + B_{_R}),\\ \bar{Qx} &= \bar{h}\frac{Qx_{_L}\sqrt{h_{_R}} + Qx_{_R}\sqrt{h_{_L}}}{h_{_L}\sqrt{h_{_R}} + h_{_R}\sqrt{h_{_L}}},\\ \bar{Qy} &= \bar{h}\frac{Qy_{_L}\sqrt{h_{_R}} + Qy_{_R}\sqrt{h_{_L}}}{h_{_L}\sqrt{h_{_R}} + h_{_R}\sqrt{h_{_L}}},\end{aligned}$$

with corresponding velocities,

$$\begin{aligned}\bar{u} &= \frac{\sqrt{h_{_R}}u_{_R} + \sqrt{h_{_L}}u_{_L}}{\sqrt{h_{_L}} + \sqrt{h_{_R}}},\\[1em] \bar{v} &= \frac{\sqrt{h_{_R}}v_{_R} + \sqrt{h_{_L}}v_{_L}}{\sqrt{h_{_L}} + \sqrt{h_{_R}}}.\end{aligned}$$

For more information on these see Hudson [34].

7.3 The 2D RKDG Method

The 2D DG method, given in [13], was applied to a scalar 2D equation. This was extended in [18] to systems of equations in 2D. Since the 2D shallow water and morphodynamical equations can be written in this form we can apply the 2D TVD RKDG method to them. We initially consider the homogeneous equations to be written in the form,

$$\frac{\partial}{\partial t}\mathbf{U} + \underline{\nabla}.\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix} = \mathbf{0},$$

with $A = \frac{\partial \mathbf{F}}{\partial \mathbf{U}}$, $B = \frac{\partial \mathbf{G}}{\partial \mathbf{U}}$ and will later extend the method to include source terms, as in (7.1).

The 2D RKDG method creates a numerical solution that is a piecewise polynomial in space and piecewise constant in time. It employs the TVD Runge-Kutta time discretisation and a discontinuous Galerkin spatial discretisation. The spatial discretisation requires a set of basis functions to span the space. We will now discuss the natural options for the choice of basis functions.

7.3.1 Basis Elements

The choice of basis elements in 2D is more complex than 1D, even when we restrict our cells to triangles. A set of linearly independent functions over a triangular cell is needed to create a basis for our numerical approximation to the solution in that cell. There are many choices of sets of basis functions, two are covered here. We consider arbitrarily shaped triangular cells of non-zero area as shown in Figure 7.1. All points are assumed to be labelled in an anti-clockwise direction.

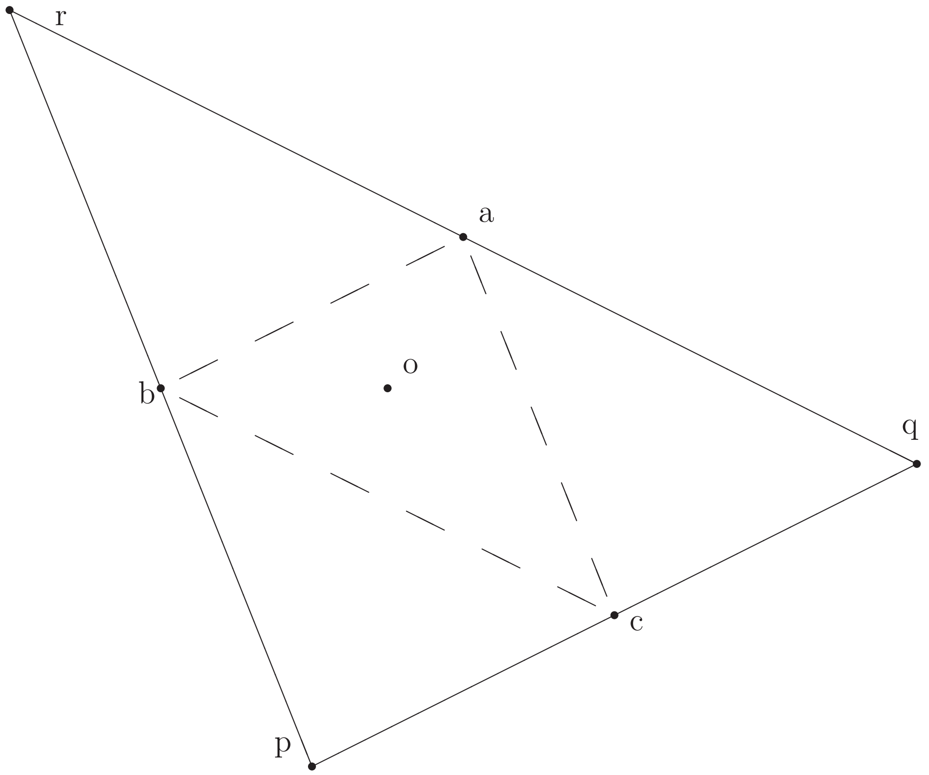


Figure 7.1: A 2D Triangular Cell

Vertex Based Basis Elements

These basis functions are equivalent to the "hat" functions in 1D. They consist of linear functions that take the value 1 at a vertex of the triangle and 0 at the other two. The solution inside the cell can then be represented as a linear combination of the values at the vertices of the cell.

Let us define a triangular cell to be represented by the three points p, q and r as shown in Figure 7.1. Then the solution can be represented as

$$\mathbf{W}(x,y,t) = \mathbf{W}_{(p)}(t)\nu_{(p)}(x,y) + \mathbf{W}_{(q)}(t)\nu_{(q)}(x,y) + \mathbf{W}_{(r)}(t)\nu_{(r)}(x,y)$$

where $\nu_{(i)}$ represents the linear basis function that takes the value 1 at vertex i and zero at the other two and $\mathbf{W}_{(i)}$ is the corresponding spatial coefficient. For example, the basis function $\nu_{(p)}$ is shown in Figure 7.2 on the left.

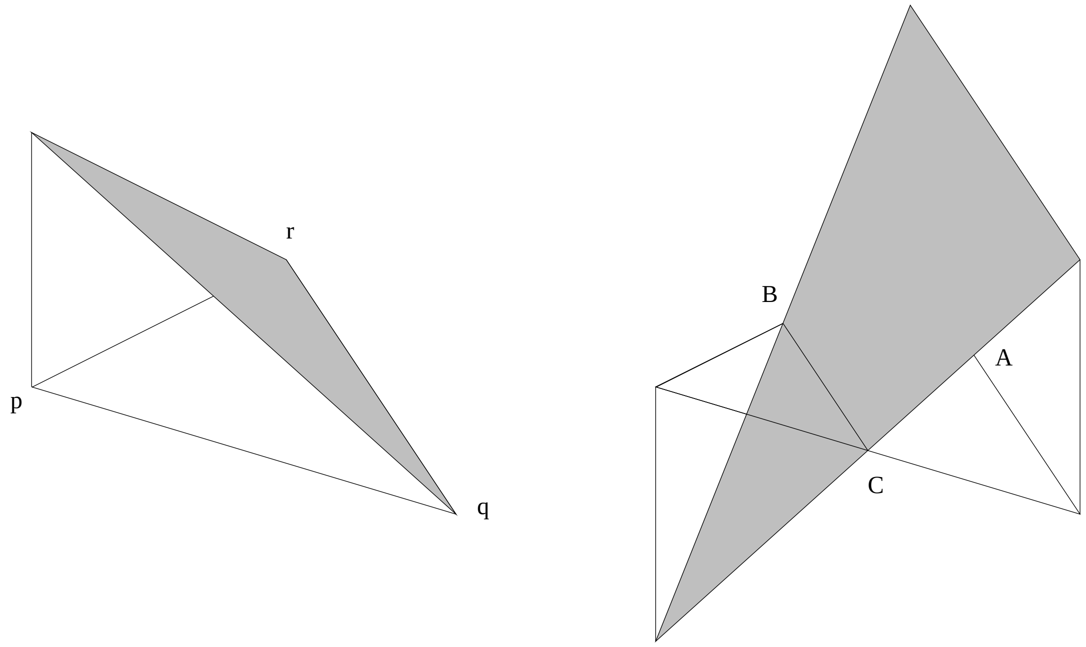


Figure 7.2: Different Basis Functions - vertex based $\nu_{(p)}$ on the left and edged based $\nu_{(a)}$ on the right.

Edge Based Basis Elements

These basis functions are equivalent to the Legendre functions in 1D. They consist of linear functions that take the value 1 along one edge of the triangle and 0 at the midpoints of the other two edges. This means that the basis function will take the value 1 at one of the midpoints of the edges and zero at the other two. Correspondingly, the values of the basis function at the cell's vertices are 1 for the two adjacent vertices and -1 at the opposing vertex. The solution can then be, simply, represented as a linear combination of the values at the mid-points of the edges of the cell. These correspond to the points a, b and c in Figure 7.1. Then the solution can be represented as

$$\mathbf{W}(x,y,t) = \mathbf{W}_{(a)}(t)\nu_{(a)}(x,y) + \mathbf{W}_{(b)}(t)\nu_{(b)}(x,y) + \mathbf{W}_{(c)}(t)\nu_{(c)}(x,y)$$

where $\nu_{(i)}$ represents the linear basis function taking the value 1 at point i and zero at the other two and $\mathbf{W}_{(i)}$ is the corresponding spatial coefficient. For example, the basis function $\nu_{(a)}$ is shown in Figure 7.2 on the right.

Area Coordinates

The definition of a basis function for an arbitrary triangle can be difficult to express. A simple method for expressing the basis function can be achieved through the use of area coordinates. Let us consider the triangle represented by the vertices p, q and r with an arbitrary point s within the triangle. This is shown in Figure 7.3. We would like to define the basis function that takes the value 1 at point p and 0 at q and r.

The basis function $\nu_{(p)}$ can be represented as the ratio of the area of the triangle $|\Delta_{sqr}|$ to that of the triangle $|\Delta_{pqr}|$,

$$\nu_{(p)} = \frac{|\Delta_{sqr}|}{|\Delta_{pqr}|}.$$

A simple definition of the area of a triangle is half the determinant of the three

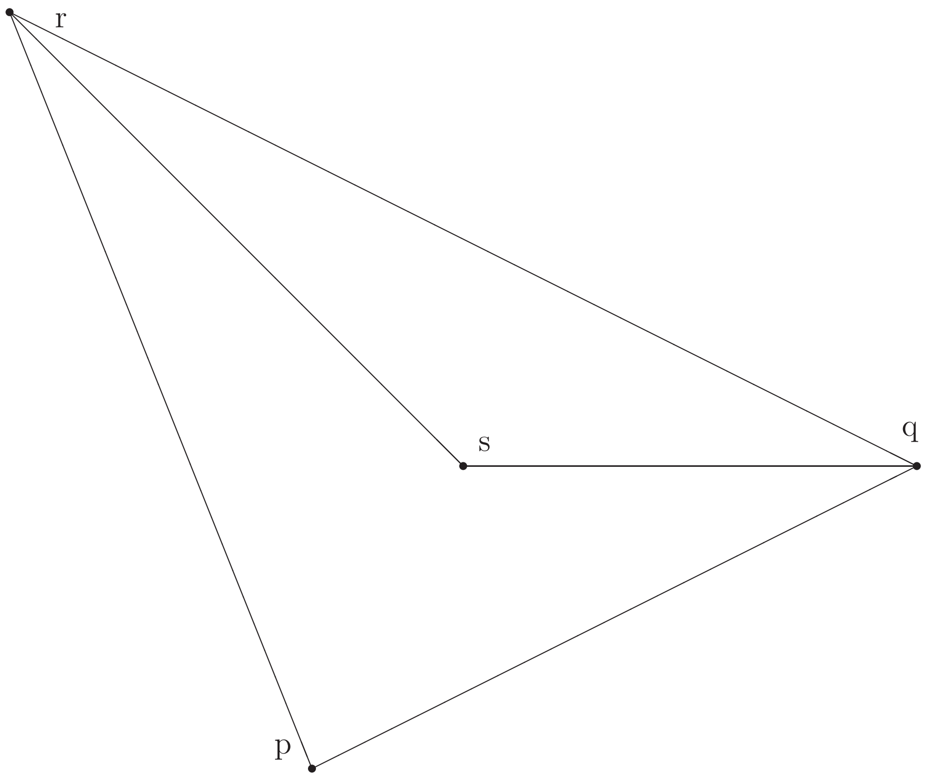


Figure 7.3: Area Coordinate Representation

dimensional coordinate matrix,

$$|\Delta_{pqr}| = {}^{1}\!/_{2}\mathrm{det}\begin{bmatrix} 1 & x_p & y_p \\ 1 & x_q & y_q \\ 1 & x_r & y_r \end{bmatrix}, \tag{7.3}$$

where the point p is at position (x_p, y_p).

With edge based basis elements the representation of the basis function is identically achieved.

The basis function $\nu_{(a)}$ can be represented as the ratio of the area of the triangle $|\Delta_{sbc}|$ to that of the triangle $|\Delta_{abc}|$,

$$\nu_{(a)} = \frac{|\Delta_{sbc}|}{|\Delta_{abc}|}.$$

This function is valid for the basis function within the entire region given by the triangle pqr not just the region given by abc. It should be noted that the sign generated via (7.3) should be maintained as this corresponds to inverting the order, clockwise and anti-clockwise, that the points are defined and this "negative area" generates a negative value of the basis function near the point opposite that which the basis function is labelled from.

Edge Normals

In the following definition, of the RKDG method, the unit normal to each edge of the triangle will be needed. As an example, the edge that contains the point a has an outward pointing normal which can be uniquely determined by the vector $\overrightarrow{qr}$. The same determination, using the vector $\overrightarrow{rq}$ should give the inward pointing normal. We can determine the outward pointing unit normal, $\mathbf{n}_a$, by

$$\overrightarrow{qr} = \begin{bmatrix} x_r - x_q \\ y_r - y_q \end{bmatrix} \qquad \mathbf{n}_a = \begin{bmatrix} y_r - y_q \\ x_q - x_r \end{bmatrix} / (\text{Length of } \overrightarrow{qr}) = \frac{N\,\overrightarrow{qr}}{\sqrt{\overrightarrow{qr}\,.\,\overrightarrow{qr}}},$$

where,

$$N = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

It is clear to see that the vector $\overrightarrow{qr}$ is identical to $2\,\overrightarrow{cb}$ thus we can write the normal in terms of the midpoints of the edges,

$$\mathbf{n}_a = 2\begin{bmatrix} y_b - y_c \\ x_c - x_b \end{bmatrix} / (\text{Length of } \overrightarrow{qr}) = \begin{bmatrix} y_b - y_c \\ x_c - x_b \end{bmatrix} / (\text{Length of } \overrightarrow{cb}). \qquad (7.4)$$

This specification of the unit normal is written entirely in terms of the solution points, thus can be coded easily.

7.3.2 Application of The DG method

The process of discretisation using the DG method is the same in 2D as was used in 1D. We will initially assume that the equations can be written in the form of a homogeneous system of hyperbolic equations,

$$\frac{\partial}{\partial t}\mathbf{U} + \underline{\nabla}. \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix} = \mathbf{0},$$

and we will extend the method to include source terms later.

In the same manner as 1D we multiply through the equation by a test function $\nu = \nu(x, y)$ and then integrate over the cell,

$$\iint_\Omega \nu \frac{\partial}{\partial t}\mathbf{U}\ d\Omega + \iint_\Omega \nu \underline{\nabla}. \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix} d\Omega = \mathbf{0}.$$

We then apply Green's theorem, otherwise known as integration by parts in 2D, to the equation to obtain,

$$\iint_\Omega \nu \frac{\partial}{\partial t}\mathbf{U}\ d\Omega + \oint_\Gamma \nu \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix} .\mathbf{n}\ d\Gamma - \iint_\Omega \mathbf{grad}\,(\nu). \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix} d\Omega = \mathbf{0},$$

where $\mathbf{n}$ is the unit outward pointing normal to the boundary of the cell.

We approximate our exact analytical solution by a polynomial representation that is continuous in a cell but we do not require the representation to be continuous across cell boundaries. This means that the representation of the solution will be discontinuous at the cell boundaries and the term within the boundary integral, the second term in the equation above, is undefined. We, therefore, replace the analytical solution $\mathbf{U}$ with the numerical approximation $\mathbf{W}$ and replace the flux functions with a numerical flux function, $\mathbf{H}$. This gives,

$$\iint_\Omega \nu \frac{\partial}{\partial t}\mathbf{W}\ d\Omega + \oint_\Gamma \nu\mathbf{H}\ d\Gamma - \iint_\Omega \mathbf{grad}\,(\nu). \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix} d\Omega = \mathbf{0}, \tag{7.5}$$

where $\mathbf{H}$ is an approximation to $[\mathbf{F}, \mathbf{G}]^T.\mathbf{n}$. We will now consider each of these terms separately.

7.3.3 The Boundary Integral Term

The second term in (7.5) is an integral around the boundary of a cell of the product of a numerical flux function and the test function. Since we are using a triangular grid we know that the boundary takes the form of three straight edges. We can see from Figure 7.1 that,

$$\oint_\Gamma \nu\mathbf{H}\ d\Gamma = \int_q^r \nu\mathbf{H}\ d\Gamma + \int_r^p \nu\mathbf{H}\ d\Gamma + \int_p^q \nu\mathbf{H}\ d\Gamma.$$

We can discretise these line integrals by using Gaussian quadrature along each edge. We choose the same quadratures as were used in the 1D case to the order necessary. In the case of a second order method, the two point Gaussian quadrature is suitable. This gives us a discretisation of

$$\begin{aligned}\oint_\Gamma \nu\mathbf{H}\, d\Gamma \quad \approx \quad & \tfrac{1}{2}|\Gamma_{qr}|\left(\nu\mathbf{H}|_{aq} + \nu\mathbf{H}|_{ar}\right)\\ & +\tfrac{1}{2}|\Gamma_{rp}|\left(\nu\mathbf{H}|_{br} + \nu\mathbf{H}|_{bp}\right)\\ & +\tfrac{1}{2}|\Gamma_{pq}|\left(\nu\mathbf{H}|_{cp} + \nu\mathbf{H}|_{cq}\right)\end{aligned}$$

where subscript aq indicates evaluating at the Gauss point between a and q, i.e. $\mathbf{a} + (1/2\sqrt{3})(\mathbf{c} - \mathbf{b})$.

We know that $\mathbf{H}$ is an approximation to $[\mathbf{F}, \mathbf{G}]^T.\mathbf{n}$ so we can consider it to be a function,

$$\mathbf{H} = \mathbf{H}(\mathbf{W}_{in}, \mathbf{W}_{out}, \mathbf{n}) \approx \begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}.\mathbf{n},$$

where $\mathbf{W}_{in}$ is the solution immediately inside the boundary of the cell and $\mathbf{W}_{out}$ is the solution immediately outside the boundary of the cell. If the boundary of the cell lies on the boundary of the domain then $\mathbf{W}_{out}$ is calculated from the boundary conditions, otherwise it is the value of the solution immediately inside the adjacent cell.

Cockburn *et al.* [18] suggests that a suitable $\mathbf{H}$ is

$$\mathbf{H} = \tfrac{1}{2}\begin{bmatrix}\mathbf{F}_{in}\\ \mathbf{G}_{in}\end{bmatrix}.\mathbf{n} + \tfrac{1}{2}\begin{bmatrix}\mathbf{F}_{out}\\ \mathbf{G}_{out}\end{bmatrix}.\mathbf{n} - \tfrac{1}{2}\alpha(\mathbf{W}_{out} - \mathbf{W}_{in}),$$

where α is a diagonal matrix with elements that are all an estimate of the absolute value of the largest eigenvalue at the boundary.

We suggest that a better numerical flux function would be,

$$\mathbf{H} = \tfrac{1}{2}\begin{bmatrix}\mathbf{F}_{in}\\ \mathbf{G}_{in}\end{bmatrix}.\mathbf{n} + \tfrac{1}{2}\begin{bmatrix}\mathbf{F}_{out}\\ \mathbf{G}_{out}\end{bmatrix}.\mathbf{n} - \tfrac{1}{2}\left|\begin{bmatrix}\mathrm{A}\\ \mathrm{B}\end{bmatrix}.\mathbf{n}\right|(\mathbf{W}_{out} - \mathbf{W}_{in}), \tag{7.6}$$

which corresponds to setting $\alpha = |[\mathrm{A}, \mathrm{B}]^T.\mathbf{n}|$. This corresponds to the first order Roe-averaged upwind numerical flux function for an arbitrary edge alignment.

The corresponding second order Roe-averaged Lax-Wendroff numerical flux function would be given by

$$\mathbf{H} = \tfrac{1}{2}\begin{bmatrix}\mathbf{F}_{in}\\ \mathbf{G}_{in}\end{bmatrix}.\mathbf{n} + \tfrac{1}{2}\begin{bmatrix}\mathbf{F}_{out}\\ \mathbf{G}_{out}\end{bmatrix}.\mathbf{n} - \tfrac{1}{2}\frac{\Delta t}{|\Gamma|}\left(\begin{bmatrix}\mathrm{A}\\ \mathrm{B}\end{bmatrix}.\mathbf{n}\right)^{2}(\mathbf{W}_{out} - \mathbf{W}_{in}). \tag{7.7}$$

To achieve these numerical fluxes we need the eigendecomposition of the matrix given by $[\mathrm{A}, \mathrm{B}]^{T}.\mathbf{n}$. The decomposition of this matrix for each formulation is given in the definition of the formulations.

7.3.4 The Domain Integral Term

The third term in (7.5) is a domain integral over the cell. Since the numerical solution is continuous within this entire cell there is no need to introduce a numerical flux function here. We can immediately see that the gradient of the test function is constant for a second order method so we can extract it from the integral,

$$\iint_{\Omega} \mathbf{grad}\,(\nu).\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix} d\Omega = \mathbf{grad}\,(\nu).\iint_{\Omega}\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix} d\Omega.$$

We can then approximate the integral using a suitable quadrature of a high enough accuracy. We choose to use quadrature which is second order accurate in both spatial dimensions,

$$\iint_{\Omega} \mathbf{grad}\,(\nu).\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix} d\Omega = \frac{|\Delta|}{60}\mathbf{grad}\,(\nu).\begin{pmatrix} 3[\mathbf{F},\mathbf{G}]_p^T + 3[\mathbf{F},\mathbf{G}]_q^T + 3[\mathbf{F},\mathbf{G}]_r^T \\ +8[\mathbf{F},\mathbf{G}]_a^T + 8[\mathbf{F},\mathbf{G}]_b^T + 8[\mathbf{F},\mathbf{G}]_c^T \\ +27[\mathbf{F},\mathbf{G}]_o^T \end{pmatrix},$$

where $|\Delta|$ is the area of the grid cell and the points of evaluation are given in Figure 7.1.

7.3.5 Time Discretisation

The first term in (7.5) is a domain integral of the product of the solution with a test function. This generates an elemental mass matrix that couples the semi-discrete equations.

Our choice of edge based basis functions generates a diagonal matrix for the mass matrix with entries of ⅓$|\Delta|$. The orthogonality that was achieved in 1D is extended to 2D using these basis functions. This means that the solutions are decoupled in space.

The second order accurate, semi-discrete equations are given by,

$$\begin{aligned}\frac{|\Delta|}{3}\frac{\partial}{\partial t}\mathbf{W}_{(i)} &= \frac{|\Delta|}{60}\mathbf{grad}\left(\nu_{(i)}\right).\left(\begin{array}{r} 3\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}_p +3\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}_q +3\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}_r \\ +8\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}_a +8\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}_b +8\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}_c \\ +27\begin{bmatrix}\mathbf{F}\\ \mathbf{G}\end{bmatrix}_o \qquad\qquad\qquad\qquad \end{array}\right) \\ &- \tfrac{1}{2}\left(\begin{array}{r} |\Gamma_{qr}|\left(\nu_{(i)}\mathbf{H}|_{aq}+\nu_{(i)}\mathbf{H}|_{ar}\right) \\ +|\Gamma_{rp}|\left(\nu_{(i)}\mathbf{H}|_{br}+\nu_{(i)}\mathbf{H}|_{bp}\right) \\ +|\Gamma_{pq}|\left(\nu_{(i)}\mathbf{H}|_{cp}+\nu_{(i)}\mathbf{H}|_{cq}\right)\end{array}\right)\end{aligned}$$

We can, again write the semi-discrete equations in the form,

$$\frac{\partial}{\partial t}\mathbf{W}_{(i)} = \mathbf{L}^h_{(i)}(\mathbf{W}).$$

To this ordinary differential equation we can apply the same TVD RK time stepping that we used in 1D, the details of which are given in Section 4.3.2.

7.4 2D Limiters

A uniformly second order accurate RKDG method will generate spurious oscillations adjacent shocks. This remains true in 2D so we would like to apply a limiter to minimise or reduce these oscillations. Cockburn *et al.* introduced a limiter in [13] but used a simpler limiter in [18]. We choose to use the latter limiter due to its ease of implementation.

We also present the MLG, Maximum Limited Gradient, limiter that is used in high order reconstruction in finite volumes and give its modification to discontinuous finite elements.

We require that any limiter is conservative in each cell and that it does not create larger slopes than the unlimited case. For the limiter to be monotonic we need only ensure that it does not create new extrema at the midpoints of the edges of the cell.

7.4.1 Cockburn's Limiter

Cockburn *et al.*, [18] ,defines two differences,

$$\tilde{W}_j = W_j - W_o \qquad \Delta W_j = \alpha_1(W_{o,j} - W_o) + \alpha_2(W_{o,k} - W_o),$$

where the point j is the midpoint of an edge and W_o is the value of the solution at the centre of the cell. $W_{o,j}$ and $W_{o,k}$ are the values of the solution at the centre of the corresponding adjacent cells. $W_{o,j}$ is given by the cell sharing the edge containing the point j and $W_{o,k}$ is chosen to ensure that α_1 and α_2 are positive. For an example refer to Figure 7.4.

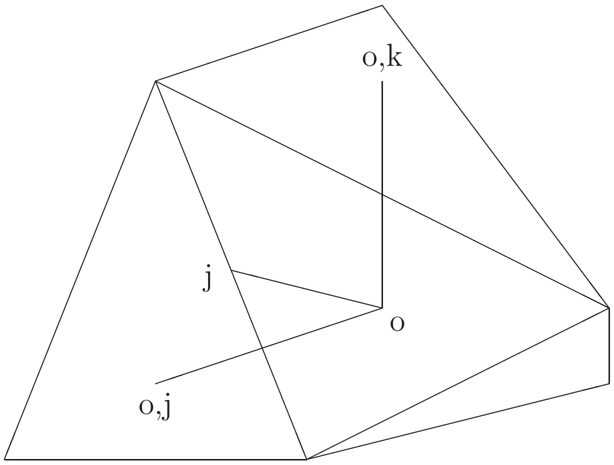


Figure 7.4: Illustration of Cockburn's Limiter

If the numerical solution was continuous with the same slope then $\tilde{W}_j = \Delta W_j$. For a discontinuous solution, however, this is not the case. We limit the slope by, for each edge, evaluating

$$\Delta_j = m(\tilde{W}_j, {}^3\!/\!_2\Delta W_j)$$

where m is the MinMod function given in Section 4.4. We choose, again, not to use the TVB version to maintain positive depth. Cockburn *et al.* choose to use the value ${}^3\!/\!_2$ in the above equation but gives no indication of the reason for this choice. We suggest that an explanation of this choice is due to the non-regularity of the grid.

For a regular grid, *i.e.* one in which the midpoints of the edges lie at the midpoint of the vector between the two midpoints of the cells, a value of 2 would be equivalent to the limiter used in 1D. For non-regular grids the value of 2 would allow spurious oscillations to be introduced and so a suitable value of ${}^3\!/\!_2$ is used to ensure this does not occur. We suggest that information contained within the values of α_1 and α_2 could determine a maximal value that ensures no oscillations are introduced into the solution.

We then determine whether $\sum \Delta_j = 0$ and, if so, set $W_j = W_o + \Delta_j$ otherwise we define

$$\begin{gathered} pos = \sum \max(\Delta_j, 0), \qquad neg = \sum \max(-\Delta_j, 0), \\ \theta^+ = \min(\frac{neg}{pos}, 1), \qquad \theta^- = \min(\frac{pos}{neg}, 1), \\ \hat{\Delta}_j = \theta^+ \max(\Delta_j, 0) - \theta^- \max(-\Delta_j, 0), \end{gathered}$$

and set $W_j = W_o + \hat{\Delta}_j$.

For systems of equations, the limiter is applied in the characteristic field. This is achieved by premultiplying the vectors $\tilde{W}_j$ and ΔW_j by the inverse eigenvector matrix to convert them into their characteristic representation. Limiting is then applied and the vector Δ_j or $\hat{\Delta}_j$ is premultiplied by the eigenvector matrix to convert back to the primitive form.

The eigenvector matrix is defined from the eigendecomposition of a linear combination of the Jacobians. For each j the combination is achieved using a direction vector aligned with j. Although [18] states that this combination is formed using the direction vector of the midpoint of the edge from the centre, correspondence with the author leads us to use the normal at the edge instead.

7.4.2 The MLG Limiter

Hubbard [32] recounts the MLG, Maximum Limited Gradient, limiter of Batten *et al.*[6] as follows.

- A gradient operator $\underline{\nabla}$ is constructed from the numerical solution to represent the slope in a cell. Hubbard chooses the midpoints of the three adjacent cells to define this gradient operator. In our case we modify the method; this slope information is already present in our solution and we can construct the gradient operator from the linear part of the representation within the cell. This eliminates the need for information from adjacent cells to construct the gradient operator.

- The limiting of the gradient operator to impose monotonicity is achieved by restricting the size of the gradient operator to the maximum that will satisfy the monotonicity conditions.

To create the gradient operator, $\underline{\nabla}$, Hubbard defines,

$$\underline{\nabla} = \begin{cases} \begin{bmatrix} -n_x/n_u \\ -n_y/n_u \end{bmatrix} & \text{for } n_u \geq \epsilon \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{otherwise,} \end{cases}$$

where $\mathbf{n} = (\mathbf{P}_i - \mathbf{P}_k) \times (\mathbf{P}_j - \mathbf{P}_k)$ and,

$$\mathbf{n}_* = \begin{bmatrix} n_x \\ n_y \\ n_u \end{bmatrix}, \qquad \mathbf{P}_* = \begin{bmatrix} x_* \\ y_* \\ u_* \end{bmatrix}.$$

Hubbard defined i, j and k to be any three solution points that are meaningful around the cell in question. Examples include any combinations of the centre points of the adjacent cells and the centre of the cell itself. For our construction we use the two combinations given by $i, j, k = a, b, c$ and the midpoints of the surrounding cells.

For each edge we calculate,

$$\alpha_* = \begin{cases} \frac{\max(u_*-u_o,0)}{\vec{r}_{*o}.\underline{\nabla}} & \text{if } \vec{r}_{*o}.\underline{\nabla} > \max(u_* - u_o, 0) \\ \frac{\min(u_*-u_o,0)}{\vec{r}_{*o}.\underline{\nabla}} & \text{if } \vec{r}_{*o}.\underline{\nabla} < \min(u_* - u_o, 0) \\ 1 & \text{otherwise,} \end{cases}$$

where $\vec{r}_{*o}$ is the vector from the centre of the cell to the midpoint of the edge and u_* is the value of the solution at the centre of the adjacent cell.

We define the MLG limiter by taking our gradient operator to be,

$$\vec{L} = \min(\alpha_a, \alpha_b, \alpha_c)\underline{\nabla},$$

and choosing the $\vec{L}$ that has greatest slope. We reconstruct our solution in the cell via,

$$u = u_o + \vec{r}\,.\,\vec{L}\,.$$

Unfortunately, we are not given specific enough instructions for performing this limiting in the characteristic field.

7.5 Boundary Conditions

As was the case with 1D, the specification of the boundary conditions can be simply expressed as the specification of the solution immediately outside the domain, $[h_{out}, Qx_{out}, Qy_{out}, B_{out}]^T$, in terms of the solution immediately inside the domain, $[h_{in}, Qx_{in}, Qy_{in}, B_{in}]^T$, and any external information we know.

7.5.1 Wall Boundaries

At a wall boundary we have a complete specification of boundary conditions given by the physical solution. At any wall we can mirror the solution. This means that we need to modify the solution normal to the boundary whilst letting the tangential solution remain unmodified. Essentially, we can mimic a wall boundary by creating a mirroring of the solution at the boundary and setting the normal component to be mirrored and the tangential component to be identical.

In the normal direction we require the surface to have zero slope and the discharge in the normal direction to be zero at the boundary. In the tangential direction we require the surface and the discharge to be identical. To achieve this we decompose the solution into its normal and tangential components. Since the depth remains identical regardless of direction we consider only the discharges. We define α_n and α_t by,

$$\begin{bmatrix} Qx_{in} \\ Qy_{in} \end{bmatrix} = \alpha_n \mathbf{n} + \alpha_t \mathbf{t}.$$

where $\mathbf{n}$ is the unit normal to the edge and $\mathbf{t}$ is the unit tangent to the edge and subscript in specifies the solution given by the numerical solution immediately inside the domain. We can then specify the solution immediately outside the domain by,

$$\begin{bmatrix} Qx_{out} \\ Qy_{out} \end{bmatrix} = -\alpha_n \mathbf{n} + \alpha_t \mathbf{t},$$

and,

$$h_{out} = h_{in} \qquad B_{out} = B_{in}.$$

7.5.2 Fluid Boundaries

With a fluid boundary we may permit water and sediment to flow through the boundary. We would like to allow information to flow out of the domain whilst only allowing information that we specify to flow into the domain. As in Section 4.6 we rely on the choice of numerical flux function to provide the well-posedness at the boundaries.

We can extend the same boundary conditions from 1D to 2D. For simple boundary conditions, as used in [13, 18], we simply set the boundary data to be the far-field data.

7.6 Source Term Discretisations

Cockburn *et al.* only covered 2D homogeneous systems of equations in the series [17, 16, 15, 13, 18] and did not provide any information about discretising source terms. We, therefore, need to find a method to discretise the source term. We seek here, as we did with 1D, a source term discretisation that satisfies the C-property without any assumption on the solution.

7.6.1 Quadrature Integration

In 1D the source term discretisation, in terms of the SWE, was defined by Schwanenberg [68] by using the same quadrature for the source term as was used for the flux term. Schwanenberg, although giving results in 2D, did not specify the source term discretisation that was used in 2D. We presume that the same principle was used and Schwanenberg used the same quadrature for the source term discretisation as was used for the flux discretisation. We shall continue to call this the FE discretisation. This means, for a second order method, our source term is approximated by,

$$\iint_{\Omega} \nu\mathbf{R}\, d\Omega \approx \frac{|\Delta|}{60}\left(\begin{array}{r} 3\nu\mathbf{R}|_p{+}3\nu\mathbf{R}|_q{+}3\nu\mathbf{R}|_r \\ +8\nu\mathbf{R}|_a{+}8\nu\mathbf{R}|_b{+}8\nu\mathbf{R}|_c \\ +27\nu\mathbf{R}|_o \qquad\qquad\qquad \end{array}\right),$$

where $\mathbf{R} = [0, -gh\frac{\partial B}{\partial x}, -gh\frac{\partial B}{\partial y}]^T$. All the derivatives $\frac{\partial B}{\partial x}$ and $\frac{\partial B}{\partial y}$ in this equation are evaluated from the inside of the cell in consideration.

We proved in Section 5.1 that, in 1D, this approach created a source term discretisation that only satisfies the C-property under the condition that the bed is continuously represented and this continues to be true in 2D.

7.6.2 A Finite Difference Discretisation

We can attempt to modify the finite difference approximation, given in 1D by (6.2) and (6.1) or (6.3) to 2D. In finite differences this is simple to achieve as finite differences assumes a rectilinear, spatially-aligned grid. In finite elements we would like a discretisation for arbitrary triangulations.

A natural place to seek evaluations is at the same points used by the boundary integral term of the flux. We can consider this to be an integral around the boundary of the cell. In total there were six points used, each at the Gauss point on each of the three edges. We therefore define the source term approximation to be,

$$\iint_{\Omega} \nu\mathbf{R}\, d\Omega \approx \begin{pmatrix} \tfrac{1}{2}|\Gamma_{qr}|\left(\nu\mathbf{S}|_{aq} + \nu\mathbf{S}|_{ar}\right) \\ +\tfrac{1}{2}|\Gamma_{rp}|\left(\nu\mathbf{S}|_{br} + \nu\mathbf{S}|_{bp}\right) \\ +\tfrac{1}{2}|\Gamma_{pq}|\left(\nu\mathbf{S}|_{cp} + \nu\mathbf{S}|_{cq}\right) \end{pmatrix}, \tag{7.8}$$

where $\mathbf{S}$ is a numerical source function dependent on the solution immediately inside the cell, the solution immediately outside the cell and the normal to the edge at the point of evaluation.

To define the evaluations we need to, firstly, consider where the A term comes from in 1D. We note that, for 1D,

$$\frac{\partial \mathbf{F}}{\partial x} - \mathbf{R} = \mathrm{A}\left(\frac{\partial \mathbf{W}}{\partial x} - \mathrm{A}^{-1}\mathbf{R}\right),$$

when $\mathbf{R} = [0, -gh\frac{\partial B}{\partial x}]^T$. We, therefore, define an approximation to the source term at any point on the boundary to be,

$$\mathbf{R}_x(\mathbf{W}_{in}, \mathbf{W}_{out}) = \begin{bmatrix} 0 \\ -g\bar{h}(B_{out} - B_{in}) \\ 0 \end{bmatrix}, \qquad \mathbf{R}_y(\mathbf{W}_{in}, \mathbf{W}_{out}) = \begin{bmatrix} 0 \\ 0 \\ -g\bar{h}(B_{out} - B_{in}) \end{bmatrix},$$

where $\bar{h}$ is given by the Roe-averages. This could create an ambiguity for an evaluation at a cell vertex so we shall avoid using vertices in the definition of the source term discretisation.

We, therefore, propose that the source term discretisation that balances the Roe-averaged upwind numerical flux, given by (7.6), is given by

$$\begin{aligned} \mathbf{S} &= \tfrac{1}{2}\left(\begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix}.\mathbf{n} - \left|\begin{bmatrix} \mathrm{A} \\ \mathrm{B} \end{bmatrix}.\mathbf{n}\right| \tfrac{1}{2}\begin{bmatrix} \mathrm{A}^{-1} \\ \mathrm{B}^{-1} \end{bmatrix}.\begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix}\right) \qquad (7.9) \\ &= \tfrac{1}{2}\left(n_x\mathbf{R}_x + n_y\mathbf{R}_y - |n_x\mathrm{A} + n_y\mathrm{B}|\,\tfrac{1}{2}\left(\mathrm{A}^{-1}\mathbf{R}_x + \mathrm{B}^{-1}\mathbf{R}_y\right)\right), \end{aligned}$$

and that the source term discretisation that balances the Roe-averaged Lax-Wendroff numerical flux, given by (7.7), is

$$\begin{aligned}\mathbf{S} &= \tfrac{1}{2}\left(\begin{bmatrix}\mathbf{R}_x\\ \mathbf{R}_y\end{bmatrix}.\mathbf{n} - \frac{\Delta t}{|\Gamma|}\left(\begin{bmatrix}\mathrm{A}\\ \mathrm{B}\end{bmatrix}.\mathbf{n}\right)^2 \tfrac{1}{2}\begin{bmatrix}\mathrm{A}^{-1}\\ \mathrm{B}^{-1}\end{bmatrix}.\begin{bmatrix}\mathbf{R}_x\\ \mathbf{R}_y\end{bmatrix}\right) \qquad (7.10)\\ &= \tfrac{1}{2}\left(n_x\mathbf{R}_x + n_y\mathbf{R}_y - \frac{\Delta t}{|\Gamma|}\left(n_x\mathrm{A} + n_y\mathrm{B}\right)^2 \tfrac{1}{2}\left(\mathrm{A}^{-1}\mathbf{R}_x + \mathrm{B}^{-1}\mathbf{R}_y\right)\right).\end{aligned}$$

We will define (7.8) along with (7.9) or (7.10) to be the FD source term discretisation.

7.6.3 The Proposed Source Term Discretisation

We propose that the superposition of the FE source term discretisation and the FD source term discretisation satisfies the C-properties without the assumptions necessary that either of them alone need to satisfy the C-property. This means that the semi-discrete form in the presence of a source term is given by,

$$\begin{aligned}\frac{|\Delta|}{3}\frac{\partial}{\partial t}\mathbf{W}_{(i)} &= \frac{|\Delta|}{60}\mathbf{grad}\left(\nu_{(i)}\right).\begin{pmatrix} 3[\mathbf{F},\mathbf{G}]_p^T + 3[\mathbf{F},\mathbf{G}]_q^T + 3[\mathbf{F},\mathbf{G}]_r^T\\ +8[\mathbf{F},\mathbf{G}]_a^T + 8[\mathbf{F},\mathbf{G}]_b^T + 8[\mathbf{F},\mathbf{G}]_c^T\\ +27[\mathbf{F},\mathbf{G}]_o^T\end{pmatrix}\\ &- \tfrac{1}{2}\begin{pmatrix} |\Gamma_{qr}|\left(\nu_{(i)}\mathbf{H}|_{aq} + \nu_{(i)}\mathbf{H}|_{ar}\right)\\ +|\Gamma_{rp}|\left(\nu_{(i)}\mathbf{H}|_{br} + \nu_{(i)}\mathbf{H}|_{bp}\right)\\ +|\Gamma_{pq}|\left(\nu_{(i)}\mathbf{H}|_{cp} + \nu_{(i)}\mathbf{H}|_{cq}\right)\end{pmatrix} \qquad (7.11)\\ &+ \frac{|\Delta|}{60}\begin{pmatrix} 3\nu_{(i)}\mathbf{R}|_p + 3\nu_{(i)}\mathbf{R}|_q + 3\nu_{(i)}\mathbf{R}|_r\\ +8\nu_{(i)}\mathbf{R}|_a + 8\nu_{(i)}\mathbf{R}|_b + 8\nu_{(i)}\mathbf{R}|_c\\ +27\nu_{(i)}\mathbf{R}|_o\end{pmatrix}\\ &+ \tfrac{1}{2}\begin{pmatrix} |\Gamma_{qr}|\left(\nu_{(i)}\mathbf{S}|_{aq} + \nu_{(i)}\mathbf{S}|_{ar}\right)\\ +|\Gamma_{rp}|\left(\nu_{(i)}\mathbf{S}|_{br} + \nu_{(i)}\mathbf{S}|_{bp}\right)\\ +|\Gamma_{pq}|\left(\nu_{(i)}\mathbf{S}|_{cp} + \nu_{(i)}\mathbf{S}|_{cq}\right)\end{pmatrix}.\end{aligned}$$

We can, again, write this in the form of an ordinary differential equation,

$$\frac{\partial}{\partial t}\mathbf{W}_{(i)} = \mathbf{L}^h_{(i)}(\mathbf{W}),$$

and apply the same TVD RK time stepping that we used in 1D, the details of which are given in Section 4.3.2.

7.7 Validation

We need to establish that the method is viable. To do this we will proceed as we did in 1D by proving that it satisfies the C-property and testing it with some test cases. To be able to prove that it satisfies the C-property we need to extend the concept of C-property to 2D.

7.7.1 The 2D C-property

The principle behind the C-property in 1D was that under some simple conditions the numerical method should accurately represent the numerical solution. These conditions were that the water should be still and the surface should be flat. This led to the requirement that, when the numerical data represented this condition, the discretisation of the flux term should balance the discretisation of the source term, creating a zero time derivative.

This can naturally be extended to 2D. For the water to be still we require,

$$u \equiv v \equiv 0,$$

and for the surface to be flat we require,

$$h = D - B,$$

where D is the constant surface elevation. For a numerical scheme to satisfy the C-property we require that under these conditions the discretisations of the flux terms should balance the discretisation of the source term.

7.7.2 C-property Proof

Before we provide the proof of C-property satisfaction we shall provide an identity that will be used in the proof. This identity will enable us to link the boundary integrals with the domain integrals. We can see that the basis function $\nu_{(a)}$ is given by

$$\nu_{(a)} = \frac{2}{|\Delta|}\det\begin{bmatrix} 1 & x & y \\ 1 & x_b & y_b \\ 1 & x_c & y_c \end{bmatrix} = \frac{2}{|\Delta|}\left(x_b y_c - x_c y_b + x(y_b - y_c) + y(x_c - x_b)\right).$$

which means that

$$\mathbf{grad}\left(\nu_{(a)}\right) = \frac{2}{|\Delta|}\begin{bmatrix} y_b - y_c \\ x_c - x_b \end{bmatrix}.$$

However, we can see, from (7.4), that

$$\mathbf{n}_a = \frac{2}{|\Gamma_{qr}|}\begin{bmatrix} y_b - y_c \\ x_c - x_b \end{bmatrix},$$

and therefore

$$|\Delta|\mathbf{grad}\left(\nu_{(a)}\right) = |\Gamma_{qr}|\mathbf{n}_a,$$

or

$$|\Delta|\frac{\partial \nu_{(a)}}{\partial x} = |\Gamma_{qr}|n_{a,x}, \qquad |\Delta|\frac{\partial \nu_{(a)}}{\partial y} = |\Gamma_{qr}|n_{a,y},$$

where $n_{a,x}$ is the component of $\mathbf{n}_a$ corresponding to the x direction. This identity can be applied with b and c to give a complete set of identities. These are,

$$\begin{array}{ll} |\Delta|\frac{\partial \nu_{(a)}}{\partial x} = |\Gamma_{qr}|n_{a,x}, & |\Delta|\frac{\partial \nu_{(a)}}{\partial y} = |\Gamma_{qr}|n_{a,y}, \\ |\Delta|\frac{\partial \nu_{(b)}}{\partial x} = |\Gamma_{rp}|n_{b,x}, & |\Delta|\frac{\partial \nu_{(b)}}{\partial y} = |\Gamma_{rp}|n_{b,y}, \\ |\Delta|\frac{\partial \nu_{(c)}}{\partial x} = |\Gamma_{pq}|n_{c,x}, & |\Delta|\frac{\partial \nu_{(c)}}{\partial y} = |\Gamma_{pq}|n_{c,y}. \end{array} \tag{7.12}$$

To prove that the method satisfies the C-property we shall consider the semi-discrete form and its application to formulation 2DSWE-C. We will use the first order Roe-averaged upwind numerical flux, (7.6), and the proposed source term discretisation using the corresponding first order Roe-averaged upwind source function given by (7.9). We shall take the requirements given by the C-property, *i.e.*,

$$h \equiv D - B, \qquad u \equiv v \equiv 0,$$

and show that the scheme reduces to $\frac{\partial \mathbf{W}}{\partial t} = 0$. To achieve this we shall rewrite the semi-discrete form, with source term discretisation, given by (7.11), as

$$\begin{aligned}\frac{|\Delta|}{3}\frac{\partial}{\partial t}\mathbf{W}_{(i)} &= \frac{|\Delta|}{60}\begin{pmatrix} 3\left(\mathbf{grad}\left(\nu_{(i)}\right).[\mathbf{F},\mathbf{G}]^T + \nu_{(i)}\mathbf{R}\right)_p \\ +3\left(\mathbf{grad}\left(\nu_{(i)}\right).[\mathbf{F},\mathbf{G}]^T + \nu_{(i)}\mathbf{R}\right)_q \\ +3\left(\mathbf{grad}\left(\nu_{(i)}\right).[\mathbf{F},\mathbf{G}]^T + \nu_{(i)}\mathbf{R}\right)_r \\ +8\left(\mathbf{grad}\left(\nu_{(i)}\right).[\mathbf{F},\mathbf{G}]^T + \nu_{(i)}\mathbf{R}\right)_a \\ +8\left(\mathbf{grad}\left(\nu_{(i)}\right).[\mathbf{F},\mathbf{G}]^T + \nu_{(i)}\mathbf{R}\right)_b \\ +8\left(\mathbf{grad}\left(\nu_{(i)}\right).[\mathbf{F},\mathbf{G}]^T + \nu_{(i)}\mathbf{R}\right)_c \\ +27\left(\mathbf{grad}\left(\nu_{(i)}\right).[\mathbf{F},\mathbf{G}]^T + \nu_{(i)}\mathbf{R}\right)_o \end{pmatrix} \\ &+ \tfrac{1}{2}\begin{pmatrix} |\Gamma_{qr}|\left(\nu_{(i)}(\mathbf{S}-\mathbf{H})|_{aq} + \nu_{(i)}(\mathbf{S}-\mathbf{H})|_{ar}\right) \\ +|\Gamma_{rp}|\left(\nu_{(i)}(\mathbf{S}-\mathbf{H})|_{br} + \nu_{(i)}(\mathbf{S}-\mathbf{H})|_{bp}\right) \\ +|\Gamma_{pq}|\left(\nu_{(i)}(\mathbf{S}-\mathbf{H})|_{cp} + \nu_{(i)}(\mathbf{S}-\mathbf{H})|_{cq}\right) \end{pmatrix}.\end{aligned} \tag{7.13}$$

This combines the source and flux discretisations into two terms. The first term in (7.13) is the domain integral of the flux and source terms and the second term is the boundary integral.

We will begin by evaluating the first term in (7.13). For any point in the cell we need to evaluate,

$$\mathbf{grad}\left(\nu_{(i)}\right).[\mathbf{F},\mathbf{G}]^T + \nu_{(i)}\mathbf{R}.$$

The C-property requires that the velocities are zero. This means that

$$\mathbf{grad}\left(\nu_{(i)}\right).[\mathbf{F},\mathbf{G}]^T + \nu_{(i)}\mathbf{R} = \begin{bmatrix} 0 \\ \tfrac{1}{2}gh^2\frac{\partial\nu_{(i)}}{\partial x} - g\nu_{(i)}h\frac{\partial B}{\partial x} \\ \tfrac{1}{2}gh^2\frac{\partial\nu_{(i)}}{\partial y} - g\nu_{(i)}h\frac{\partial B}{\partial y} \end{bmatrix}.$$

The C-property also requires that $B \equiv D - h$ and since D is constant this also means that,

$$\frac{\partial B}{\partial x} = -\frac{\partial h}{\partial x}, \qquad \frac{\partial B}{\partial y} = -\frac{\partial h}{\partial y},$$

giving,

$$\mathbf{grad}\left(\nu_{(i)}\right).[\mathbf{F},\mathbf{G}]^T + \nu_{(i)}\mathbf{R} = \begin{bmatrix} 0 \\ \tfrac{1}{2}gh^2\frac{\partial\nu_{(i)}}{\partial x} + gh\nu_{(i)}\frac{\partial h}{\partial x} \\ \tfrac{1}{2}gh^2\frac{\partial\nu_{(i)}}{\partial y} + gh\nu_{(i)}\frac{\partial h}{\partial y} \end{bmatrix}.$$

Inserting this into the first term in (7.13) gives

$$\frac{g|\Delta|}{60}\left(\begin{array}{lll} 3\begin{bmatrix}0\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial x}+h\nu_{(i)}\frac{\partial h}{\partial x}\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial y}+h\nu_{(i)}\frac{\partial h}{\partial y}\end{bmatrix}_p & +3\begin{bmatrix}0\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial x}+h\nu_{(i)}\frac{\partial h}{\partial x}\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial y}+h\nu_{(i)}\frac{\partial h}{\partial y}\end{bmatrix}_q & +3\begin{bmatrix}0\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial x}+h\nu_{(i)}\frac{\partial h}{\partial x}\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial y}+h\nu_{(i)}\frac{\partial h}{\partial y}\end{bmatrix}_r \\ +8\begin{bmatrix}0\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial x}+h\nu_{(i)}\frac{\partial h}{\partial x}\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial y}+h\nu_{(i)}\frac{\partial h}{\partial y}\end{bmatrix}_a & +8\begin{bmatrix}0\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial x}+h\nu_{(i)}\frac{\partial h}{\partial x}\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial y}+h\nu_{(i)}\frac{\partial h}{\partial y}\end{bmatrix}_b & +8\begin{bmatrix}0\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial x}+h\nu_{(i)}\frac{\partial h}{\partial x}\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial y}+h\nu_{(i)}\frac{\partial h}{\partial y}\end{bmatrix}_c \\ +27\begin{bmatrix}0\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial x}+h\nu_{(i)}\frac{\partial h}{\partial x}\\ \frac{1}{2}h^2\frac{\partial\nu_{(i)}}{\partial y}+h\nu_{(i)}\frac{\partial h}{\partial y}\end{bmatrix}_o & & \end{array}\right),$$

which can be rewritten as,

$$\frac{g|\Delta|}{60}\frac{1}{2}\begin{pmatrix}3{h_p}^2+3{h_q}^2+3{h_r}^2\\ +8{h_a}^2+8{h_b}^2+8{h_c}^2\\ +27{h_o}^2\end{pmatrix}\begin{bmatrix}0\\ \frac{\partial\nu_{(i)}}{\partial x}\\ \frac{\partial\nu_{(i)}}{\partial y}\end{bmatrix}+\frac{g|\Delta|}{60}\begin{pmatrix}3h_p\nu_{(i),p}+3h_q\nu_{(i),q}+3h_r\nu_{(i),r}\\ +8h_a\nu_{(i),a}+8h_b\nu_{(i),b}+8h_c\nu_{(i),c}\\ +27h_o\nu_{(i),o}\end{pmatrix}\begin{bmatrix}0\\ \frac{\partial h}{\partial x}\\ \frac{\partial h}{\partial y}\end{bmatrix},$$

where $\nu_{(i),j}$ indicates the test function $\nu_{(i)}$ evaluated at the point j. This can be simplified, by rewriting p, q, r and o in terms of a, b and c, to,

$$\frac{g|\Delta|}{3}\frac{1}{2}\left({h_a}^2+{h_b}^2+{h_c}^2\right)\begin{bmatrix}0\\ \frac{\partial\nu_{(i)}}{\partial x}\\ \frac{\partial\nu_{(i)}}{\partial y}\end{bmatrix}+\frac{g|\Delta|}{3}\left(h_a\nu_{(i),a}+h_b\nu_{(i),b}+h_c\nu_{(i),c}\right)\begin{bmatrix}0\\ \frac{\partial h}{\partial x}\\ \frac{\partial h}{\partial y}\end{bmatrix}. \tag{7.14}$$

We will now turn our attention to the second term in (7.13). Let us now consider any point on the boundary. The we need to evaluate $\mathbf{S}-\mathbf{H}$ at this point. The numerical flux function, $\mathbf{H}$, as given by (7.6), is

$$\mathbf{H}=\frac{1}{2}\begin{bmatrix}\mathbf{F}_{in}\\ \mathbf{G}_{in}\end{bmatrix}.\mathbf{n}+\frac{1}{2}\begin{bmatrix}\mathbf{F}_{out}\\ \mathbf{G}_{out}\end{bmatrix}.\mathbf{n}-\frac{1}{2}\left|\begin{bmatrix}\mathrm{A}\\ \mathrm{B}\end{bmatrix}.\mathbf{n}\right|(\mathbf{W}_{out}-\mathbf{W}_{in}).$$

The C-property requires that the velocities are zero. Referring to (7.2), this makes the numerical flux function

$$\mathbf{H} \;=\; \frac{1}{2}\begin{bmatrix}0\\ \frac{1}{2}n_x g({h_{in}}^2+{h_{out}}^2)\\ \frac{1}{2}n_y g({h_{in}}^2+{h_{out}}^2)\end{bmatrix}-\frac{1}{2}\tilde{c}\begin{bmatrix}1 & 0 & 0\\ 0 & {n_x}^2 & n_x n_y\\ 0 & n_x n_y & {n_x}^2\end{bmatrix}\begin{bmatrix}h_{out}-h_{in}\\ 0\\ 0\end{bmatrix}$$

$$= {}^1\!/_2 \begin{bmatrix} -\sqrt{{}^1\!/_2 g(h_{in}+h_{out})}(h_{out}-h_{in}) \\ {}^1\!/_2 n_x g({h_{in}}^2+{h_{out}}^2) \\ {}^1\!/_2 n_y g({h_{in}}^2+{h_{out}}^2) \end{bmatrix}.$$

The numerical source function, as given by (7.9), is

$$\mathbf{S} = {}^1\!/_2 \left(\begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix} .\mathbf{n} - \left| \begin{bmatrix} \mathrm{A} \\ \mathrm{B} \end{bmatrix} .\mathbf{n} \right| {}^1\!/_2 \begin{bmatrix} \mathrm{A}^{-1} \\ \mathrm{B}^{-1} \end{bmatrix} . \begin{bmatrix} \mathbf{R}_x \\ \mathbf{R}_y \end{bmatrix} \right).$$

The C-property requires that the velocities are zero. Referring to (7.2), this makes the numerical source function,

$$\begin{aligned} \mathbf{S} &= {}^1\!/_2 \begin{bmatrix} 0 \\ -n_x g\tilde{h}(B_{out}-B_{in}) \\ 0 \end{bmatrix} + {}^1\!/_2 \begin{bmatrix} 0 \\ 0 \\ -n_y g\tilde{h}(B_{out}-B_{in}) \end{bmatrix} \\ &\quad -{}^1\!/_2\sqrt{g\tilde{h}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & {n_x}^2 & n_x n_y \\ 0 & n_x n_y & {n_x}^2 \end{bmatrix} {}^1\!/_2 \begin{bmatrix} 0 & 1/g\tilde{h} & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ -g\tilde{h}(B_{out}-B_{in}) \\ 0 \end{bmatrix} \\ &\quad -{}^1\!/_2\sqrt{g\tilde{h}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & {n_x}^2 & n_x n_y \\ 0 & n_x n_y & {n_x}^2 \end{bmatrix} {}^1\!/_2 \begin{bmatrix} 0 & 0 & 1/g\tilde{h} \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -g\tilde{h}(B_{out}-B_{in}) \end{bmatrix} \\ &= {}^1\!/_2 \begin{bmatrix} \sqrt{g\tilde{h}}(B_{out}-B_{in}) \\ -n_x g\tilde{h}(B_{out}-B_{in}) \\ -n_y g\tilde{h}(B_{out}-B_{in}) \end{bmatrix}. \end{aligned}$$

The C-property also requires that $B \equiv D - h$. This means that we can replace $B_{out} - B_{in}$ with $h_{in} - h_{out}$ giving,

$$\begin{aligned} \mathbf{S} &= {}^1\!/_2 \begin{bmatrix} -\sqrt{{}^1\!/_2 g(h_{in}+h_{out})}(h_{out}-h_{in}) \\ {}^1\!/_2 n_x g(h_{in}+h_{out})(h_{out}-h_{in}) \\ {}^1\!/_2 n_y g(h_{in}+h_{out})(h_{out}-h_{in}) \end{bmatrix} \\ &= {}^1\!/_2 \begin{bmatrix} -\sqrt{{}^1\!/_2 g(h_{in}+h_{out})}(h_{out}-h_{in}) \\ {}^1\!/_2 n_x g({h_{out}}^2-{h_{in}}^2) \\ {}^1\!/_2 n_y g({h_{out}}^2-{h_{in}}^2) \end{bmatrix}. \end{aligned}$$

Therefore,

$$\begin{aligned}\mathbf{S}-\mathbf{H} &= {}^1\!/\!_2\begin{bmatrix} -\sqrt{{}^1\!/\!_2 g(h_{in}+h_{out})}(h_{out}-h_{in}) \\ {}^1\!/\!_2 n_x g({h_{out}}^2-{h_{in}}^2) \\ {}^1\!/\!_2 n_y g({h_{out}}^2-{h_{in}}^2)\end{bmatrix} - {}^1\!/\!_2\begin{bmatrix} -\sqrt{{}^1\!/\!_2 g(h_{in}+h_{out})}(h_{out}-h_{in}) \\ {}^1\!/\!_2 n_x g({h_{in}}^2+{h_{out}}^2) \\ {}^1\!/\!_2 n_y g({h_{in}}^2+{h_{out}}^2)\end{bmatrix} \\ &= -{}^1\!/\!_2 g{h_{in}}^2\begin{bmatrix}0\\ n_x\\ n_y\end{bmatrix}. \qquad (7.15)\end{aligned}$$

This shows that, under the requirements of the C-property, any dependence on adjacent cells is automatically balanced in the evaluation of the numerical flux and source functions. As an additional note, if the scheme uses the second order Roe-averages Lax-Wendroff numerical flux function, given by (7.7), and the corresponding numerical source function, given by (7.10), then (7.15) still applies and the proof is the same from this point.

When we insert (7.15) into the second term in (7.13) we get

$$\begin{aligned}&{}^1\!/\!_2|\Gamma_{qr}|\left(-{}^1\!/\!_2 g\nu_{(i),aq}{h_{aq}}^2\begin{bmatrix}0\\ n_{a,x}\\ n_{a,y}\end{bmatrix} - {}^1\!/\!_2 g\nu_{(i),ar}{h_{ar}}^2\begin{bmatrix}0\\ n_{a,x}\\ n_{a,y}\end{bmatrix}\right)\\ &+{}^1\!/\!_2|\Gamma_{rp}|\left(-{}^1\!/\!_2 g\nu_{(i),br}{h_{br}}^2\begin{bmatrix}0\\ n_{b,x}\\ n_{b,y}\end{bmatrix} - {}^1\!/\!_2 g\nu_{(i),bp}{h_{bp}}^2\begin{bmatrix}0\\ n_{b,x}\\ n_{b,y}\end{bmatrix}\right)\\ &+{}^1\!/\!_2|\Gamma_{pq}|\left(-{}^1\!/\!_2 g\nu_{(i),cp}{h_{cp}}^2\begin{bmatrix}0\\ n_{c,x}\\ n_{c,y}\end{bmatrix} - {}^1\!/\!_2 g\nu_{(i),cq}{h_{cq}}^2\begin{bmatrix}0\\ n_{c,x}\\ n_{c,y}\end{bmatrix}\right),\end{aligned}$$

where $\nu_{(i),aq}$ indicates the test function $\nu_{(i)}$ evaluated at the Gauss point between a

and q and correspondingly for the others. This can be rewritten as

$$\begin{array}{c} -{}^1\!/_4 g|\Gamma_{qr}|\left(\nu_{(i),aq}{h_{aq}}^2+\nu_{(i),ar}{h_{ar}}^2\right)\begin{bmatrix}0\\ n_{a,x}\\ n_{a,y}\end{bmatrix}\\ -{}^1\!/_4 g|\Gamma_{rp}|\left(\nu_{(i),br}{h_{br}}^2+\nu_{(i),bp}{h_{bp}}^2\right)\begin{bmatrix}0\\ n_{b,x}\\ n_{b,y}\end{bmatrix}\\ -{}^1\!/_4 g|\Gamma_{pq}|\left(\nu_{(i),cp}{h_{cp}}^2+\nu_{(i),cq}{h_{cq}}^2\right)\begin{bmatrix}0\\ n_{c,x}\\ n_{c,y}\end{bmatrix}\end{array}.$$

Inserting the identities given by (7.12) into this gives,

$$\begin{array}{c} -{}^1\!/_4 g|\Delta|\left(\nu_{(i),aq}{h_{aq}}^2+\nu_{(i),ar}{h_{ar}}^2\right)\begin{bmatrix}0\\ \frac{\partial\nu_{(a)}}{\partial x}\\ \frac{\partial\nu_{(a)}}{\partial y}\end{bmatrix}\\ -{}^1\!/_4 g|\Delta|\left(\nu_{(i),br}{h_{br}}^2+\nu_{(i),bp}{h_{bp}}^2\right)\begin{bmatrix}0\\ \frac{\partial\nu_{(b)}}{\partial x}\\ \frac{\partial\nu_{(b)}}{\partial y}\end{bmatrix}\\ -{}^1\!/_4 g|\Delta|\left(\nu_{(i),cp}{h_{cp}}^2+\nu_{(i),cq}{h_{cq}}^2\right)\begin{bmatrix}0\\ \frac{\partial\nu_{(c)}}{\partial x}\\ \frac{\partial\nu_{(c)}}{\partial y}\end{bmatrix}\end{array}. \tag{7.16}$$

Therefore (7.13) is, by combining (7.14) and (7.16), equivalent to,

$$\begin{aligned}\frac{|\Delta|}{3}\frac{\partial}{\partial t}\mathbf{W}_{(i)} &= \frac{g|\Delta|}{3}{}^1\!/_2\left({h_a}^2+{h_b}^2+{h_c}^2\right)\begin{bmatrix}0\\ \frac{\partial\nu_{(i)}}{\partial x}\\ \frac{\partial\nu_{(i)}}{\partial y}\end{bmatrix}\\ &\quad +\frac{g|\Delta|}{3}\left(h_a\nu_{(i),a}+h_b\nu_{(i),b}+h_c\nu_{(i),c}\right)\begin{bmatrix}0\\ \frac{\partial h}{\partial x}\\ \frac{\partial h}{\partial y}\end{bmatrix}\end{aligned}$$

$$
\begin{aligned}
&-{}^1\!/\!_4 g|\Delta|\left(\nu_{(i),aq}{h_{aq}}^2+\nu_{(i),ar}{h_{ar}}^2\right)\begin{bmatrix}0\\ \frac{\partial\nu_{(a)}}{\partial x}\\ \frac{\partial\nu_{(a)}}{\partial y}\end{bmatrix}\\
&-{}^1\!/\!_4 g|\Delta|\left(\nu_{(i),br}{h_{br}}^2+\nu_{(i),bp}{h_{bp}}^2\right)\begin{bmatrix}0\\ \frac{\partial\nu_{(b)}}{\partial x}\\ \frac{\partial\nu_{(b)}}{\partial y}\end{bmatrix}\\
&-{}^1\!/\!_4 g|\Delta|\left(\nu_{(i),cp}{h_{cp}}^2+\nu_{(i),cq}{h_{cq}}^2\right)\begin{bmatrix}0\\ \frac{\partial\nu_{(c)}}{\partial x}\\ \frac{\partial\nu_{(c)}}{\partial y}\end{bmatrix}.
\end{aligned}
$$

We can see that the first elements of this vector equation are zero and that the second and third elements differ only by the direction in which the derivative is evaluated. It is clear to see that if we can prove a balance in one direction then it will automatically apply in the other direction. We therefore define ∂ to indicate $\frac{\partial}{\partial x}$ or $\frac{\partial}{\partial y}$, divide through by the value $g|\Delta|/3$ and consider one element of this vector equation. This gives,

$$
\begin{aligned}
\frac{1}{g}\frac{\partial}{\partial t}W_{(i)} &= {}^1\!/\!_2\left({h_a}^2+{h_b}^2+{h_c}^2\right)\partial\nu_{(i)}\\
&\quad+\left(h_a\nu_{(i),a}+h_b\nu_{(i),b}+h_c\nu_{(i),c}\right)\partial h\\
&\quad-\frac{3}{4}\left(\nu_{(i),aq}{h_{aq}}^2+\nu_{(i),ar}{h_{ar}}^2\right)\partial\nu_{(a)}\\
&\quad-\frac{3}{4}\left(\nu_{(i),br}{h_{br}}^2+\nu_{(i),bp}{h_{bp}}^2\right)\partial\nu_{(b)}\\
&\quad-\frac{3}{4}\left(\nu_{(i),cp}{h_{cp}}^2+\nu_{(i),cq}{h_{cq}}^2\right)\partial\nu_{(c)}.
\end{aligned}
$$

The proof, so far, has been independent of test function choice. Therefore, the above equation applies for all three of our test functions. If we can prove that the scheme satisfies the C-property for one test function then it is a simple process to prove that it also satisfies the C-property for the other two. We arbitrarily choose $\nu_{(i)}=\nu_{(a)}$ to demonstrate this proof with the process equally applying to $\nu_{(i)}=\nu_{(b)}$ or $\nu_{(i)}=\nu_{(c)}$.

With $\nu_{(i)} = \nu_{(a)}$ the above equation becomes,

$$\begin{aligned}
\frac{1}{g}\frac{\partial}{\partial t}W_{(a)} &= {}^1\!/_2\left({h_a}^2 + {h_b}^2 + {h_c}^2\right)\partial\nu_{(a)} \\
&\quad + \left(h_a\nu_{(a),a} + h_b\nu_{(a),b} + h_c\nu_{(a),c}\right)\partial h \\
&\quad - \frac{3}{4}\left(\nu_{(a),aq}{h_{aq}}^2 + \nu_{(a),ar}{h_{ar}}^2\right)\partial\nu_{(a)} \\
&\quad - \frac{3}{4}\left(\nu_{(a),br}{h_{br}}^2 + \nu_{(a),bp}{h_{bp}}^2\right)\partial\nu_{(b)} \\
&\quad - \frac{3}{4}\left(\nu_{(a),cp}{h_{cp}}^2 + \nu_{(a),cq}{h_{cq}}^2\right)\partial\nu_{(c)}.
\end{aligned}$$

We can evaluate the values of $\nu_{(a)}$ at the points of interest to give,

$$\begin{aligned}
\frac{1}{g}\frac{\partial}{\partial t}W_{(a)} &= {}^1\!/_2\left({h_a}^2 + {h_b}^2 + {h_c}^2\right)\partial\nu_{(a)} \\
&\quad + h_a\partial h \\
&\quad - \frac{3}{4}\left({h_{aq}}^2 + {h_{ar}}^2\right)\partial\nu_{(a)} \\
&\quad - \frac{3}{4}\left(\frac{1}{\sqrt{3}}{h_{br}}^2 - \frac{1}{\sqrt{3}}{h_{bp}}^2\right)\partial\nu_{(b)} \\
&\quad - \frac{3}{4}\left(-\frac{1}{\sqrt{3}}{h_{cp}}^2 + \frac{1}{\sqrt{3}}{h_{cq}}^2\right)\partial\nu_{(c)}.
\end{aligned}$$

Rewriting the Gauss points in terms of a, b and c and expanding ∂h gives,

$$\begin{aligned}
\frac{1}{g}\frac{\partial}{\partial t}W_{(a)} &= {}^1\!/_2\left({h_a}^2 + {h_b}^2 + {h_c}^2\right)\partial\nu_{(a)} \\
&\quad + h_a(h_a\partial\nu_{(a)} + h_b\partial\nu_{(b)} + h_c\partial\nu_{(c)}) \\
&\quad - {}^1\!/_2\left(3{h_a}^2 + (h_b - h_c)^2\right)\partial\nu_{(a)} \\
&\quad - \left(h_a h_b - h_b h_c\right)\partial\nu_{(b)} \\
&\quad - \left(h_a h_c - h_b h_c\right)\partial\nu_{(c)}.
\end{aligned}$$

Collecting in terms of ∂ gives

$$\begin{aligned}
\frac{1}{g}\frac{\partial}{\partial t}W_{(a)} &= {}^1\!/_2\left({h_a}^2 + {h_b}^2 + {h_c}^2 + 2{h_a}^2 - 3{h_a}^2 - (h_b - h_c)^2\right)\partial\nu_{(a)} \\
&\quad + \left(h_a h_b - h_a h_b + h_b h_c\right)\partial\nu_{(b)} \\
&\quad + \left(h_a h_c - h_a h_c + h_b h_c\right)\partial\nu_{(c)} \\
&= \left(h_b h_c\right)\partial\nu_{(a)}
\end{aligned}$$

$$
\begin{aligned}
& + (h_b h_c)\, \partial \nu_{(b)} \\
& + (h_b h_c)\, \partial \nu_{(c)} \\
= \;& h_b h_c (\partial \nu_{(a)} + \partial \nu_{(b)} + \partial \nu_{(c)}) \\
= \;& 0.
\end{aligned}
$$

Therefore when $\nu_{(i)} = \nu_{(a)}$ the semi-discrete form of the equations, under the requirements of the C-property, reduces to, $\frac{\partial}{\partial t}\mathbf{W}_{(a)} = \mathbf{0}$, and by similar argument, $\frac{\partial}{\partial t}\mathbf{W}_{(b)} = \mathbf{0}$, and $\frac{\partial}{\partial t}\mathbf{W}_{(c)} = \mathbf{0}$.

This means that the scheme, with the proposed source term discretisation, satisfies the conditions required for satisfaction of the C-property. We can be assured that the scheme will settle to the physically correct steady state in the presence of quiescent flow.

7.7.3 Test Cases

Having proved that the method with the proposed source term discretisation satisfies the C-property we will now demonstrate the method with numerical test cases. We will provide test cases to demonstrate the C-property satisfaction and the performance of the scheme for morphodynamical modelling.

For Test Case A and Test Case B we shall use a standard uniform 50x50 grid triangulated with alternating diagonals and assume that the domain of interest is a square region of size $1km$ square. For Test Case C we will use a standard uniform 100x8 grid triangulated with alternating diagonals and assume that the region of interest is a region of size $1km$ by $80m$. Although the tests given here will be given their dimensionalised form we will use the non-dimensionalised form for the calculation and redimensionalise the results to display them. We will also use adaptive time stepping with the wave speeds calculated from both Jacobians.

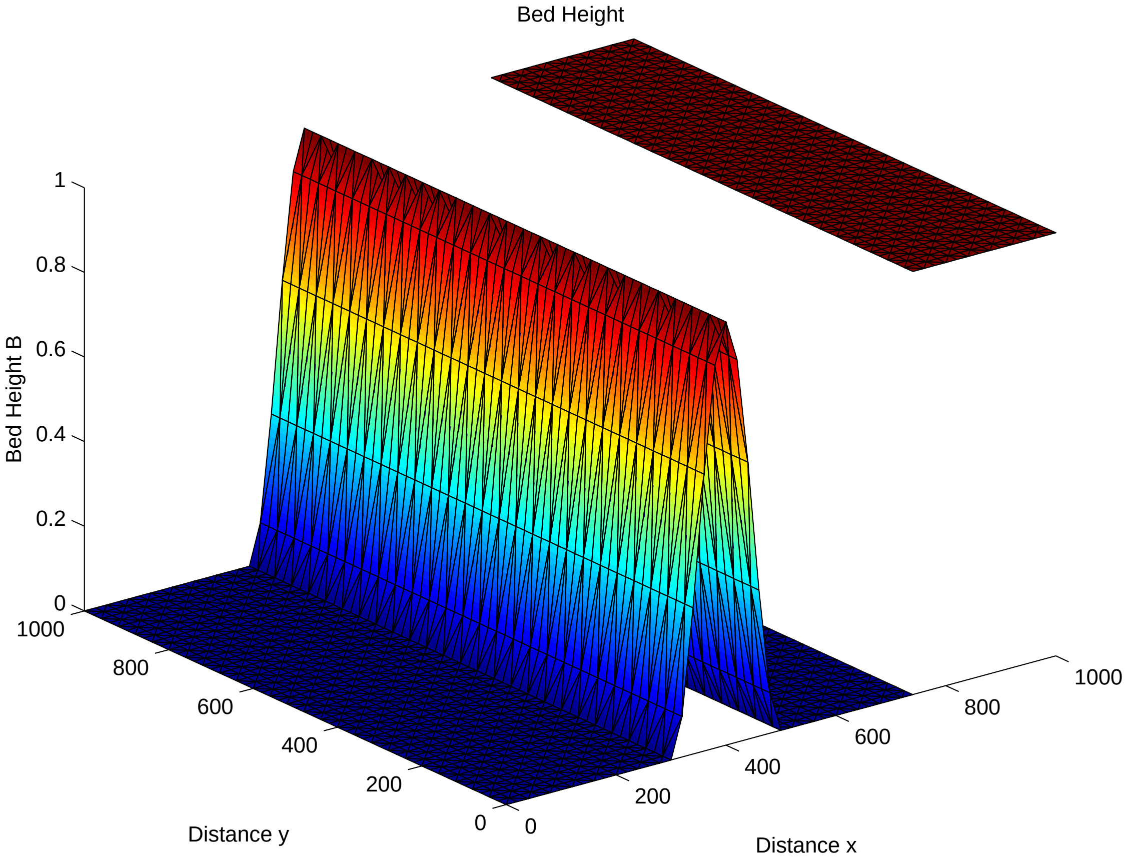


Figure 7.5: The Bed Profile For Test Case A

7.7.4 Test Case A

The simplest method for generating test cases is to extend the 1D test cases to 2D. We will therefore define test case A in 2D to be identical to test case A in 1D with the additional requirement that $v \equiv 0$ initially and on the boundary. This test case will verify that the scheme satisfies the C-property requirements. Any deviation of v from 0 will indicate a numerical error and can be a measure of the performance of the scheme.

The initial data is given by,

$$
\begin{aligned}
u(x,y,0) &= 0, \\
v(x,y,0) &= 0, \\
h(x,y,0) &= 10 - B(x,y,0), \\
B(x,y,t) &= \begin{cases} 0, & \text{if } \quad 0 \le x \le \ 300 \\ \sin^2\left(\Pi \frac{(x-300)}{200}\right), & \text{if } 300 \le x \le \ 500 \\ 0, & \text{if } 500 \le x < \ 750 \\ 1, & \text{if } 750 \le x \le 1000 \end{cases} .
\end{aligned}
$$

The bed profile for this test case is given in Figure 7.5.

The boundary data for $x = 0$ and $x = 1000$ is given by,

$$
\left.\begin{array}{ll} h(x,y,t) = 10, & B(x,y,t) = 0 \\ u(x,y,t) = 0, & v(x,y,t) = 0 \end{array}\right\}, \text{ for } |x - 500| = 500.
$$

For the boundaries at $y = 0$ and $y = 1000$ we will define wall boundaries to retain the flow inside the domain.

7.7.5 Test Case B

Test case A essentially reduced the 2D problem to 1D. We also need to verify that the method performs well in a truly 2D case. In test case B we will define a truly 2D bed profile. This will verify that the scheme satisfies the C-property independently of direction.

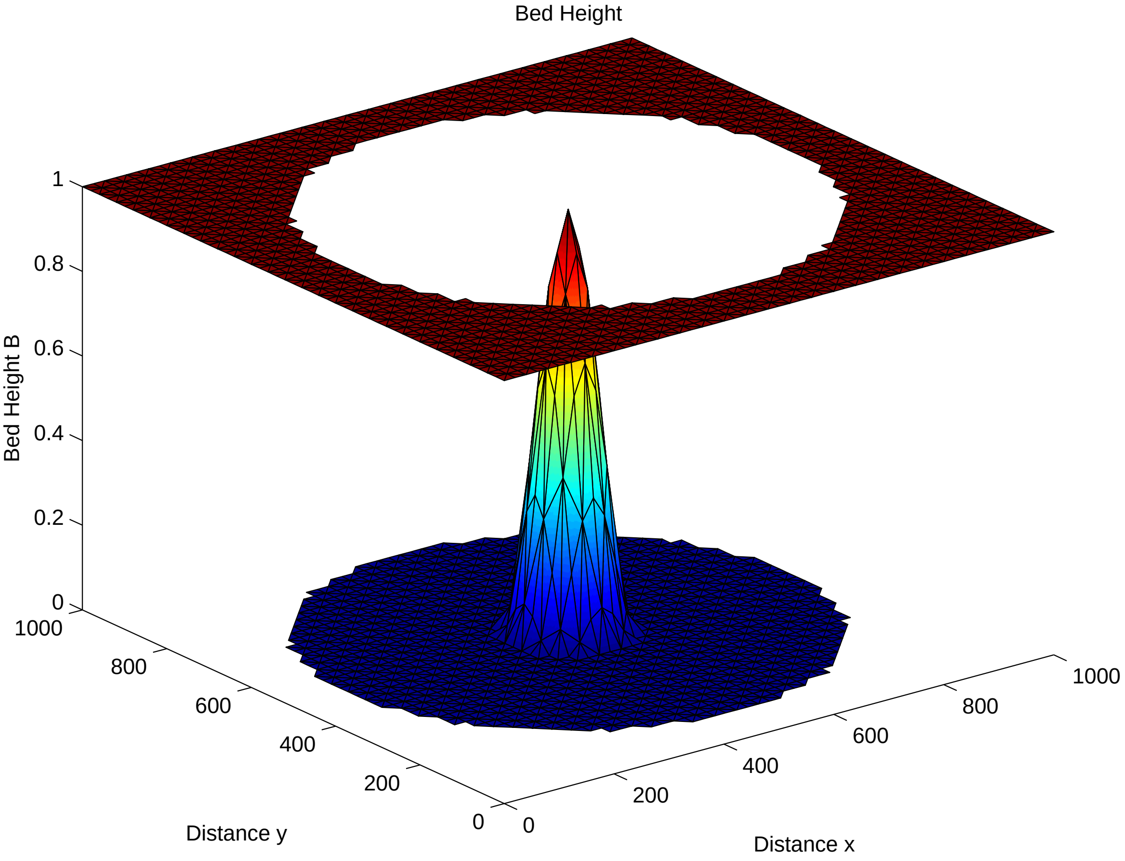


Figure 7.6: The Bed Profile For Test Case B

The initial data is given by,

$$\begin{aligned} u(x,y,0) &= 0, \\ v(x,y,0) &= 0, \\ h(x,y,0) &= 10 - B(x,y,0). \end{aligned}$$

For the bed profile we shall define,

$$B(x,y,t) = \cos^2\left(\Pi\frac{D(x,y)}{200}\right), \text{ if } D(x,y) < 100,$$

where $D(x,y) = \sqrt{(x-500)^2 + (y-500)^2}$. For the rest of the domain we will set the bed in the entire cell to 1 if the midpoint of the cell satisfies $D(x,y) > 400$ and 0 otherwise. The bed profile for this test case is given in Figure 7.6.

The boundary data is given by,

$$\left.\begin{array}{ll} h(x,y,t) = 9, & B(x,y,t) = 1 \\ u(x,y,t) = 0, & v(x,y,t) = 0 \end{array}\right\}, \text{ for } |x-500| = 500 \text{ or } |y-500| = 500.$$

7.7.6 Test Case C

The simplest method for generating test cases is to extend the 1D test cases to 2D. We will therefore define Test Case C in 2D to be identical to Test Case B in 1D with the additional requirement that $v{\equiv}0$ initially and on flow boundaries. Any deviation of v from 0 will indicate a numerical error and can be a measure of the performance of the scheme. This will allow us to test the morphodynamics in 2D with the advantage of being able to compare to the results given in 1D. We will, again, run the test for a preliminary time of $1000s$ with a fixed bed to allow the water to settle to a steady state and avoid an impulsive start.

The initial data is given by,

$$\begin{aligned} u(x,y,-1000) &= 1, \\ v(x,y,-1000) &= 0, \\ h(x,y,-1000) &= 10 - B(x,y,-1000), \end{aligned}$$

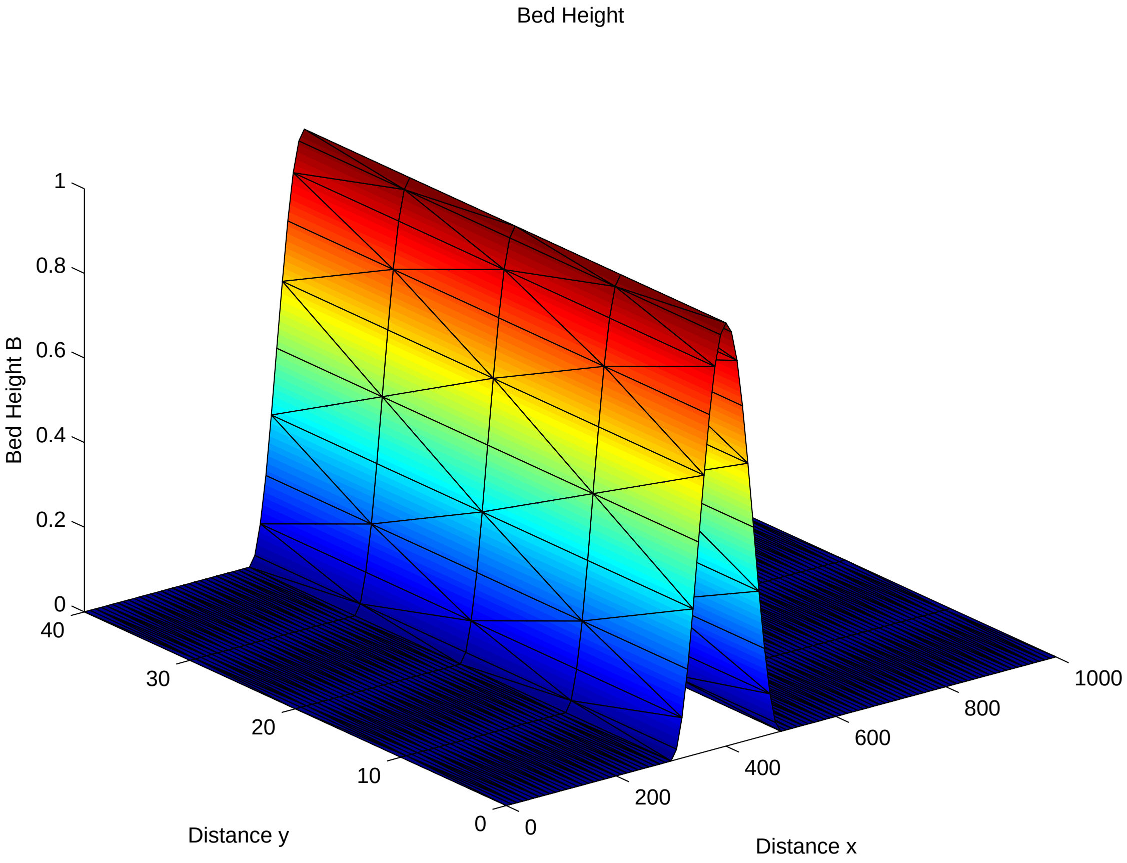


Figure 7.7: The Bed Profile For Test Case C

$$B(x,y,-1000) \quad = \quad \begin{cases} 0, & \text{if} \quad 0 \le x \le \ 300 \\ \sin^2\left(\Pi \frac{(x-300)}{200}\right), & \text{if } 300 \le x \le \ 500 \\ 0, & \text{if } 500 \le x \le 1000 \end{cases} .$$

The bed profile for this test case is given in Figure 7.7.

The boundary data for $x = 0$ and $x = 1000$ is given by,

$$\left. \begin{array}{ll} h(x,y,t) = 10, & B(x,y,t) = 0 \\ u(x,y,t) = 1, & v(x,y,t) = 0 \end{array} \right\}, \text{ for } |x - 500| = 500.$$

For the boundaries at $y = 0$ and $y = 80$ we will define wall boundaries to retain the flow inside the domain. We will initially maintain a fixed bed and then let it evolve by setting,

$$A = \begin{cases} 0, & \text{if } t < 0 \\ 0.001, & \text{if } 0 \le t \end{cases} .$$

7.7.7 Results

To demonstrate the capabilities of the method, when combined with the proposed source term discretisation and the two speed time stepping, we will show the results for when the FE discretisation, the FD discretisation and the proposed discretisation are used.

7.7.8 Results for Test Case A

Figure 7.8 shows the surface of the steady state solution when using the FE source discretisation. It is clear that the method has correctly modelled the smoothly represented bump but has failed to model the discontinuous step on the right. The result of this is a uniform depth of $9.5m$ in the entire region except over the bump. This depth is due to the difference in the boundary conditions. One boundary condition states that the depth should be $9m$ and the other states that it should be $10m$. Since the method does not see the step in the bed it assumes that there must be flow in from one side and out of the other. The steady state solution actually has a velocity of $1ms^{-1}$ in the x direction.

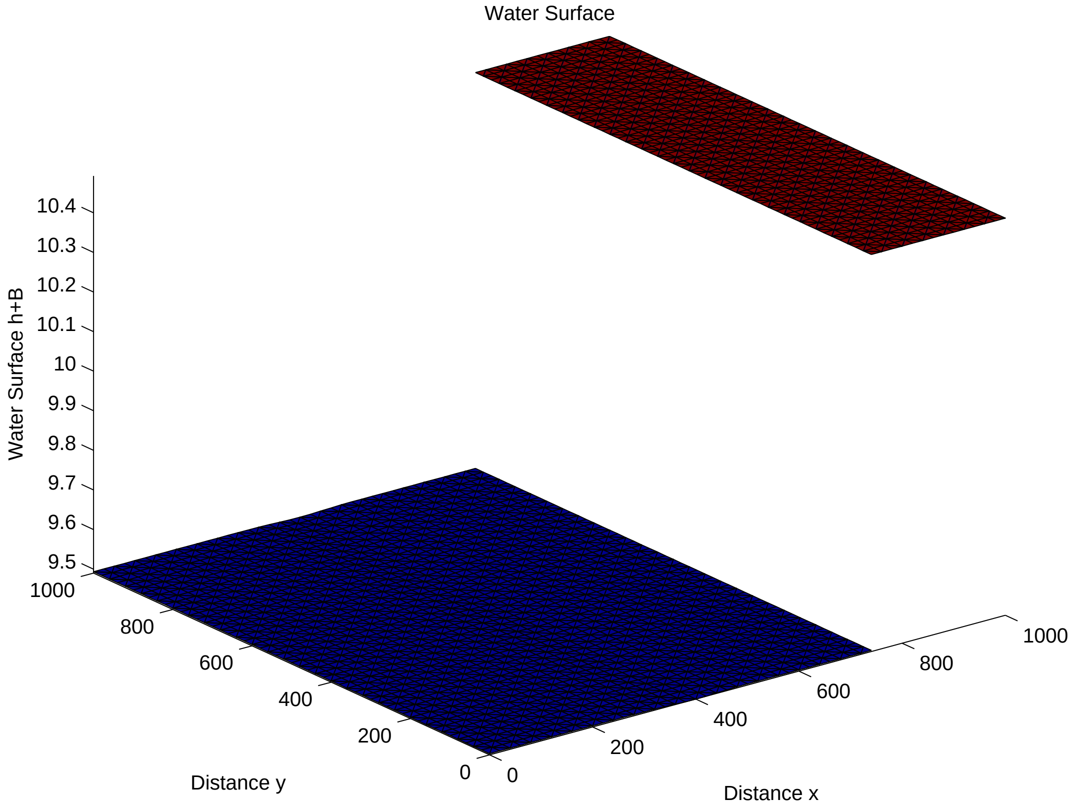


Figure 7.8: Test Case A Results with FE Source Discretisation Showing the Scheme is not C-property Satisfying

Figure 7.9 shows the surface of the steady state solution when using the FD source discretisation. It is clear that the method can correctly model the discontinuous step on the right of the region but fails to see the smoothly represented bump on the left. The result of this is a uniform surface height of $10m$ everywhere but over the bump.

Figure 7.10 shows the surface of the steady state solution when using the proposed source discretisation. It is clear that the method can correctly model the entire solution, with the only deviation from the analytical solution due to computational error. The variation in colour indicates the computational error which is of the order 10^{-15}.

7.7.9 Results for Test Case B

Figure 7.11 shows the surface of the steady state solution when using the FE source discretisation, use by Schwanenberg [68]. It is clear that the method has correctly modelled the smoothly represented bump in the middle but has failed to model the discontinuous jump around it. The result of this is a uniform depth, except over the bump, of $9m$ due to the boundary conditions.

Figure 7.12 shows the surface of the steady state solution when using the FD source discretisation. It is clear that the method can correctly model the discontinuous step that borders the region but fails to see the smoothly represented bump in the middle. The result of this is a uniform surface height of $10m$ everywhere except over the bump.

Figure 7.13 shows the surface of the steady state solution when using the proposed source discretisation. It is clear that the method can correctly model the entire solution, with the only deviation from the analytical solution due to computational error. The variation in colour indicates the computational error which is of the order 10^{-15}.

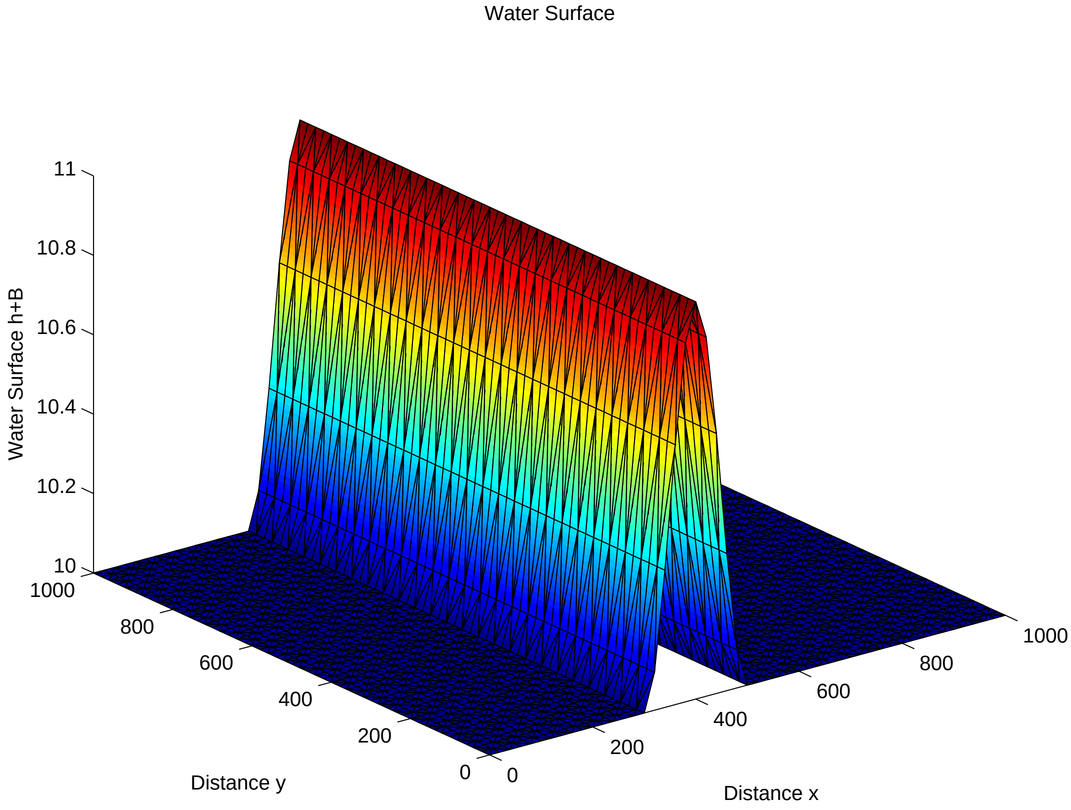


Figure 7.9: Test Case A Results with FD Source Discretisation Showing the Scheme is not C-property Satisfying

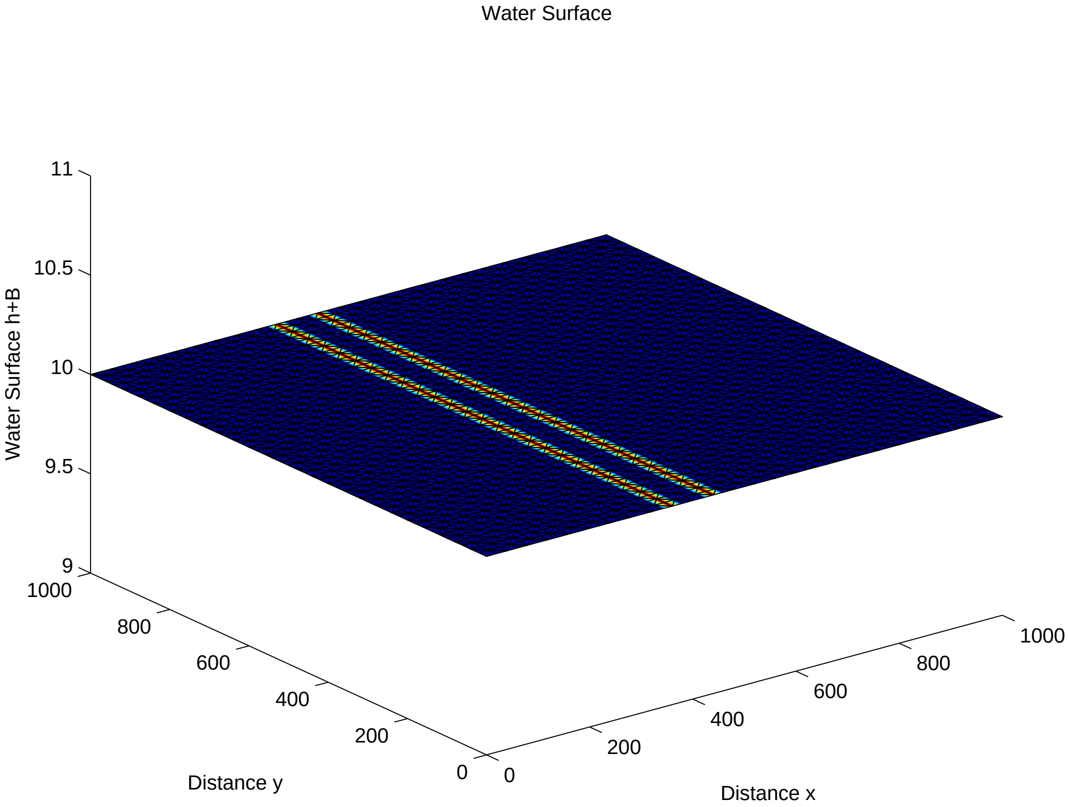


Figure 7.10: Test Case A Results with the Proposed Source Discretisation Demonstrating C-property Satisfaction

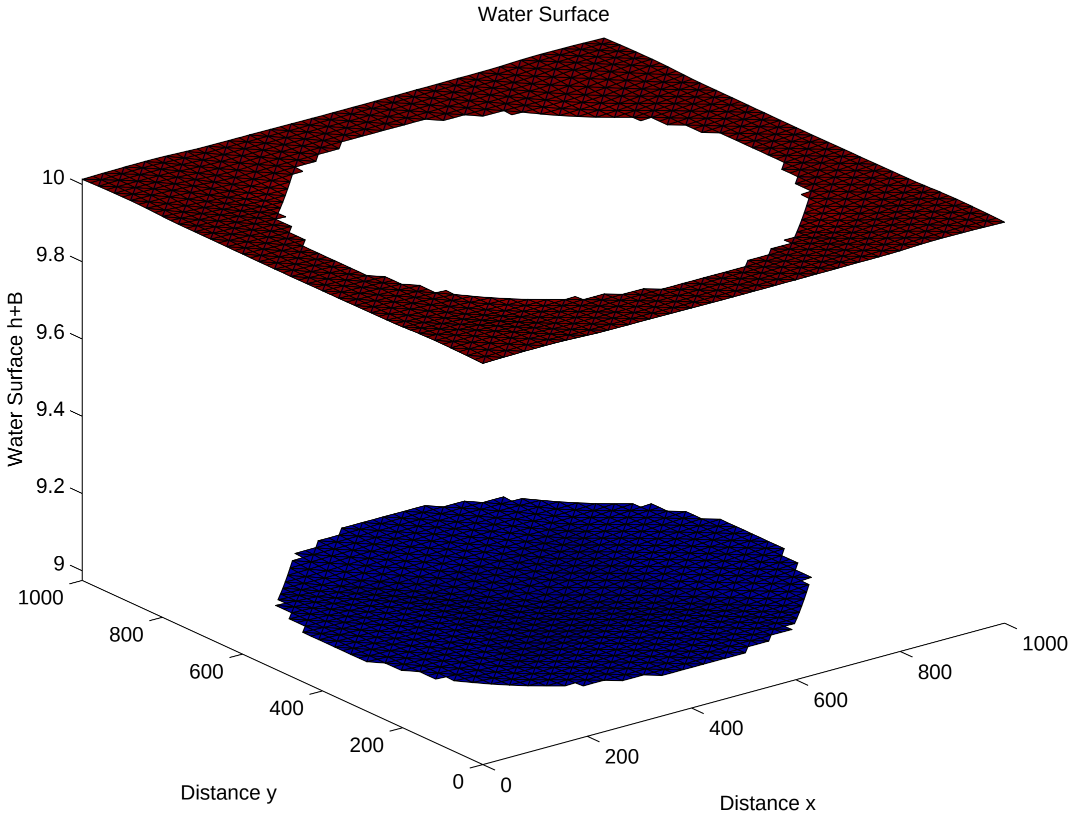


Figure 7.11: Test Case B Results with FE Source Discretisation Showing the Scheme is not C-property Satisfying

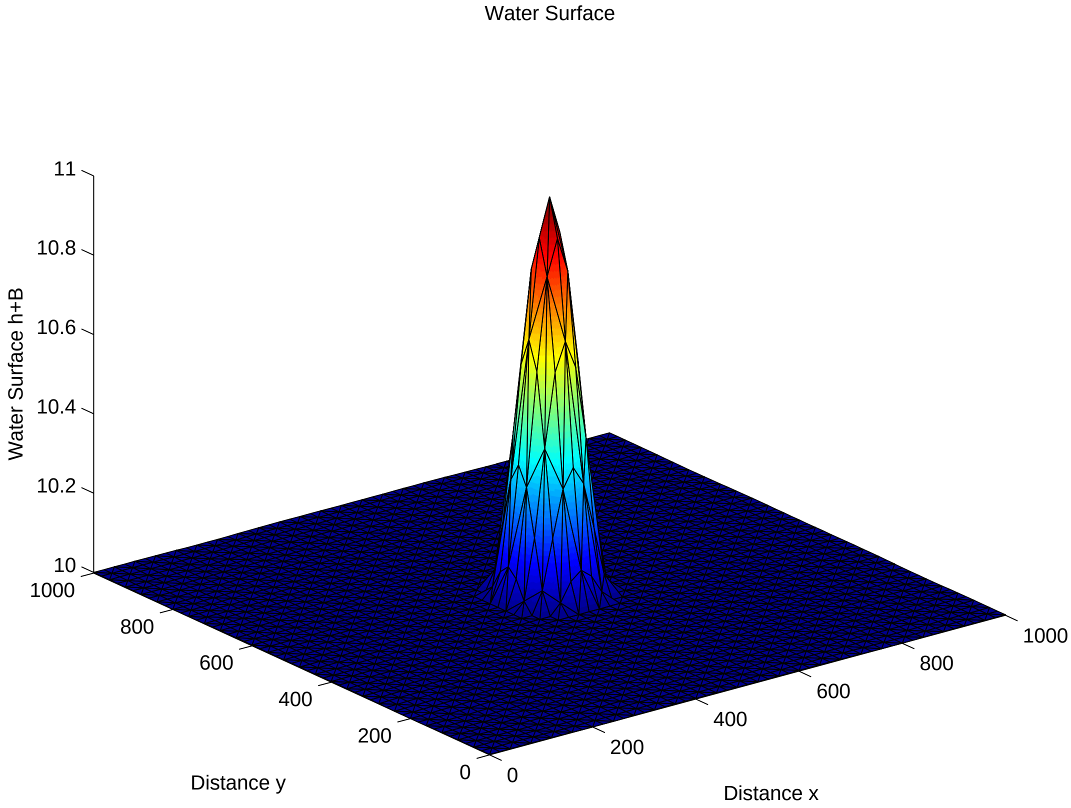


Figure 7.12: Test Case B Results with FD Source Discretisation Showing the Scheme is not C-property Satisfying

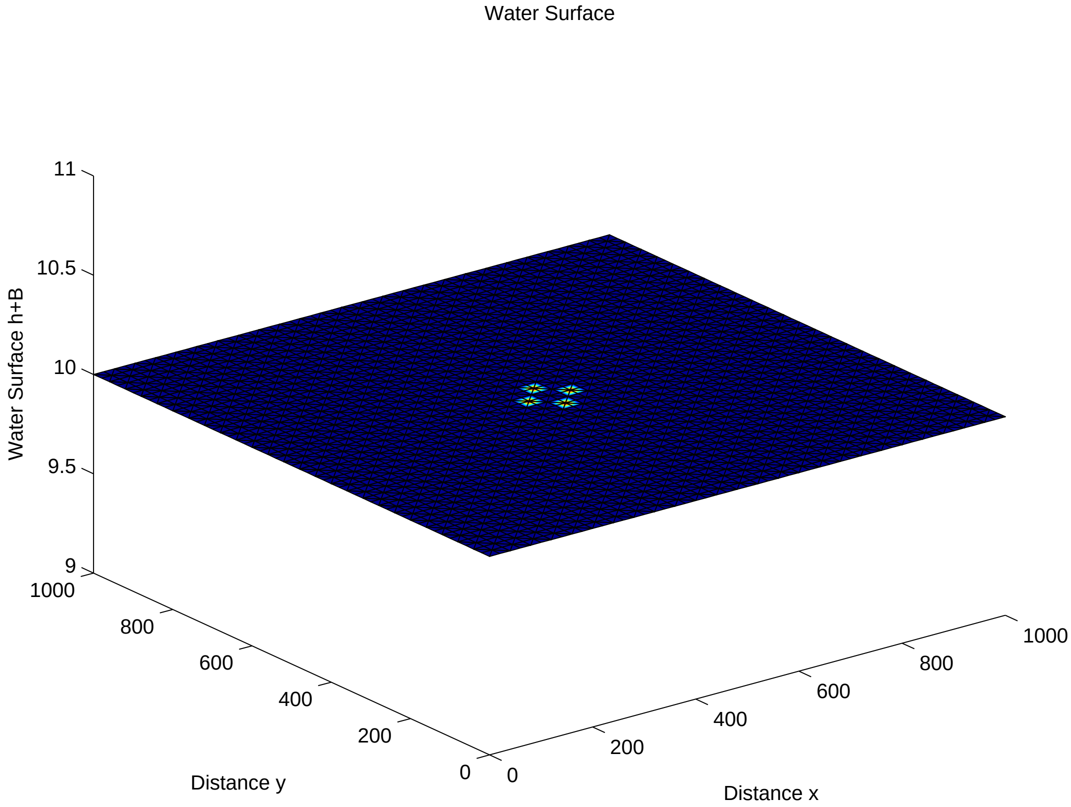


Figure 7.13: Test Case B Results with the Proposed Source Discretisation Demonstrating C-property Satisfaction

7.7.10 Results for Test Case C

Figure 7.14, Figure 7.15 and Figure 7.16 give the results of Test Case C using formulation SPLIT-C with no limiter. Figure 7.17, Figure 7.18 and Figure 7.19 give the results for the MLG limiter. The y-axis has been stretched so that the cells are more visible. The middle and bottom graphs show the same results from different views to aid understanding.

We can see that both, the unlimited and limited, methods give a good definition of the bump in the bed with an appropriate water surface profile. We see some spatial difference in the y direction for the Order 2D case only and this can be explained through the reduced number of possible planes usable for the MLG limiter. In this calculation we chose to exclude any possible plane that used values on the boundary however we recognise that we could have used these boundary values without any problems. We still see small oscillations in the solution despite using a limiter and these are an artifact of not being able to limit in the characteristic field.

Figure 7.20 gives the shape of the surface and bed profile in a side view. This allows us to compare 1D and 2D results by comparing these graphs to those in Figure 6.19. We can see that the depth of the bed bump, the depth of the surface bump and the positions of both of these correspond to the 1D case indicating that 1D motion has been captured correctly.

It should be noted that the RKDG method gives a piecewise polynomial representation in each cell and this is what has been displayed in the graphs. As the method can utilise discontinuities this may appear to make the solution appear worse than recognised finite volume schemes which are typically displayed with the mean in each cell connected to form the plane. Figure 7.21 shows different angles of the same unlimited and limited results for the bed but with the mean in each cell connected in the same fashion as is typically used with finite volume schemes. We can see that the results “appear” much better despite being generated from exactly the same data.

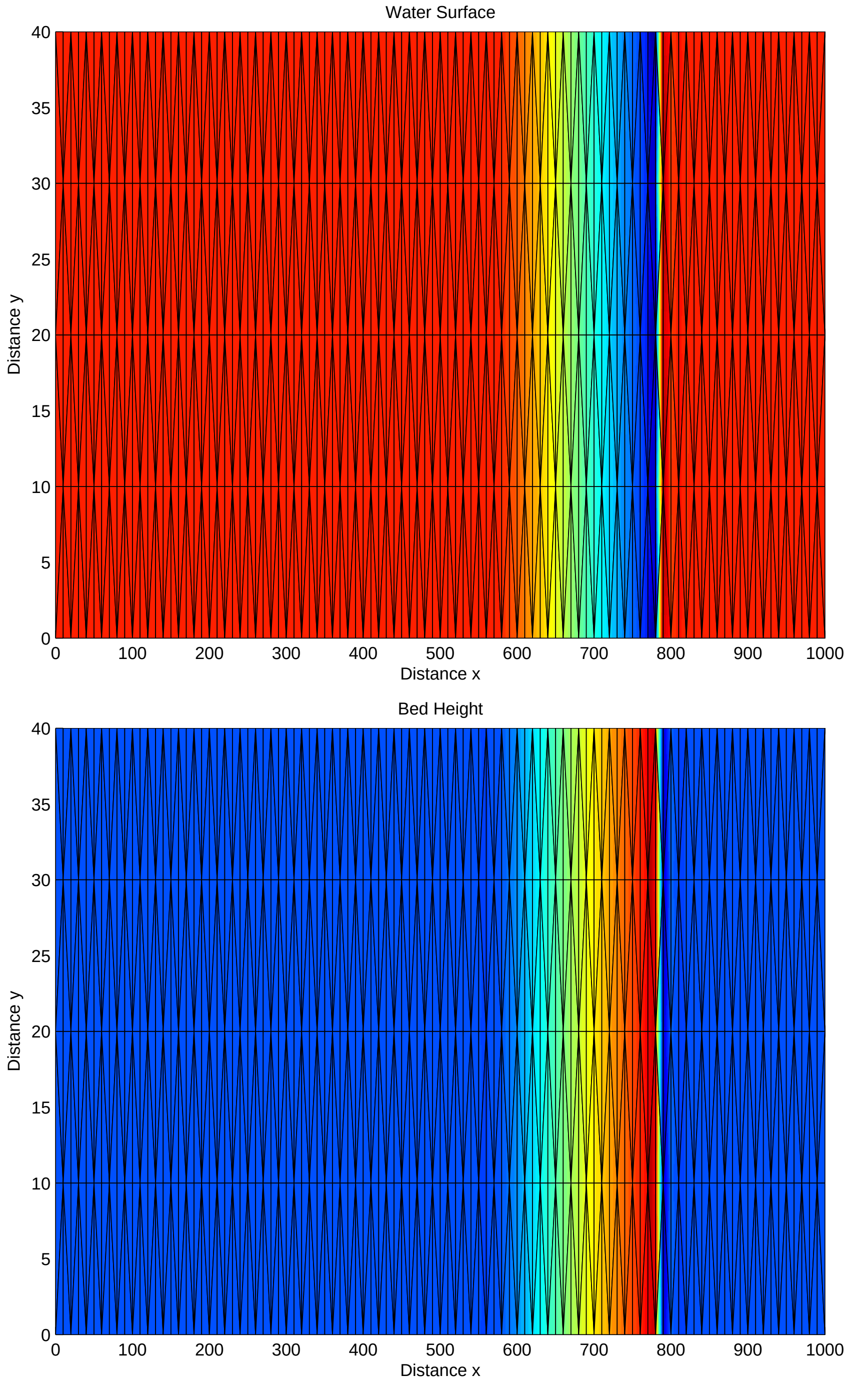


Figure 7.14: 2D Test Case C Results with the Proposed Source Discretisation for Formulation SPLIT-CB Order 2U

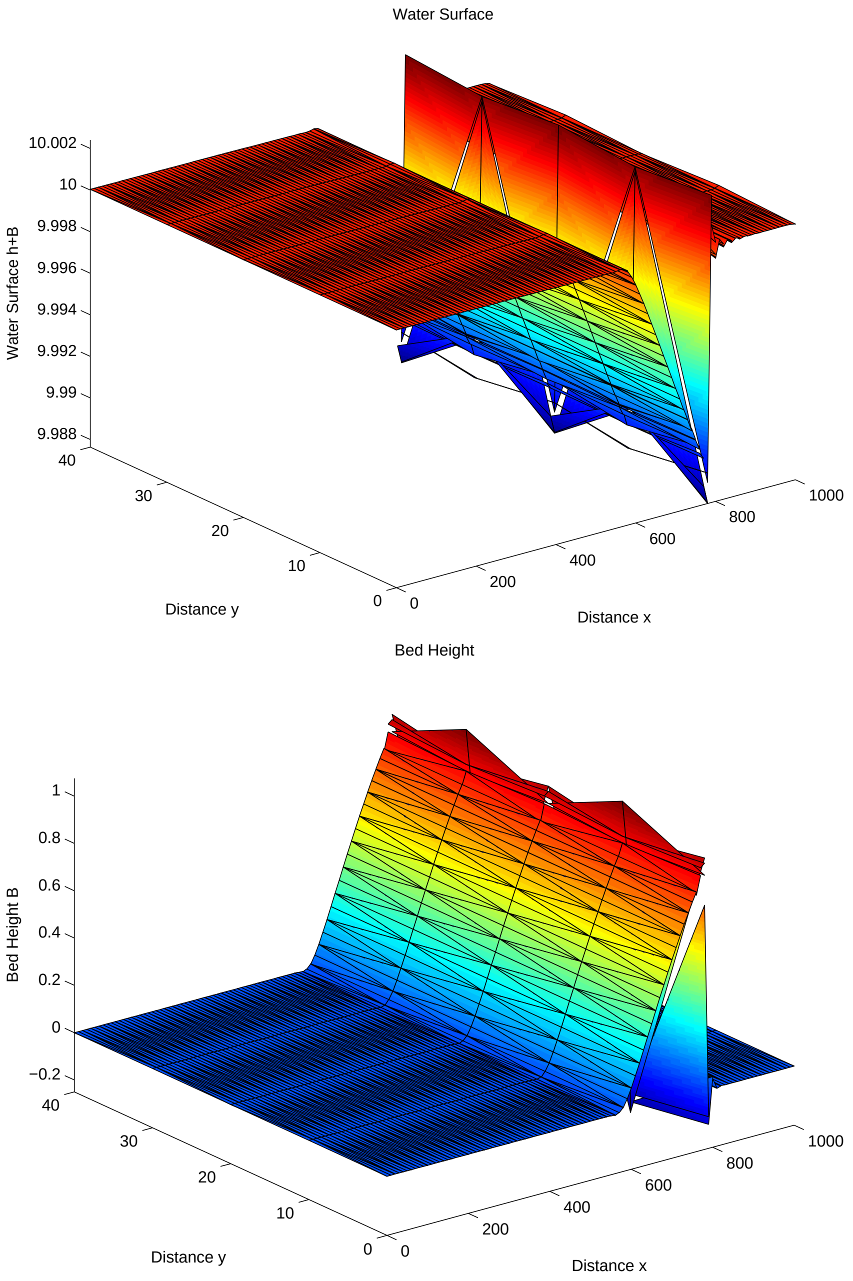


Figure 7.15: 3D Test Case C Results with the Proposed Source Discretisation for Formulation SPLIT-CB Order 2U

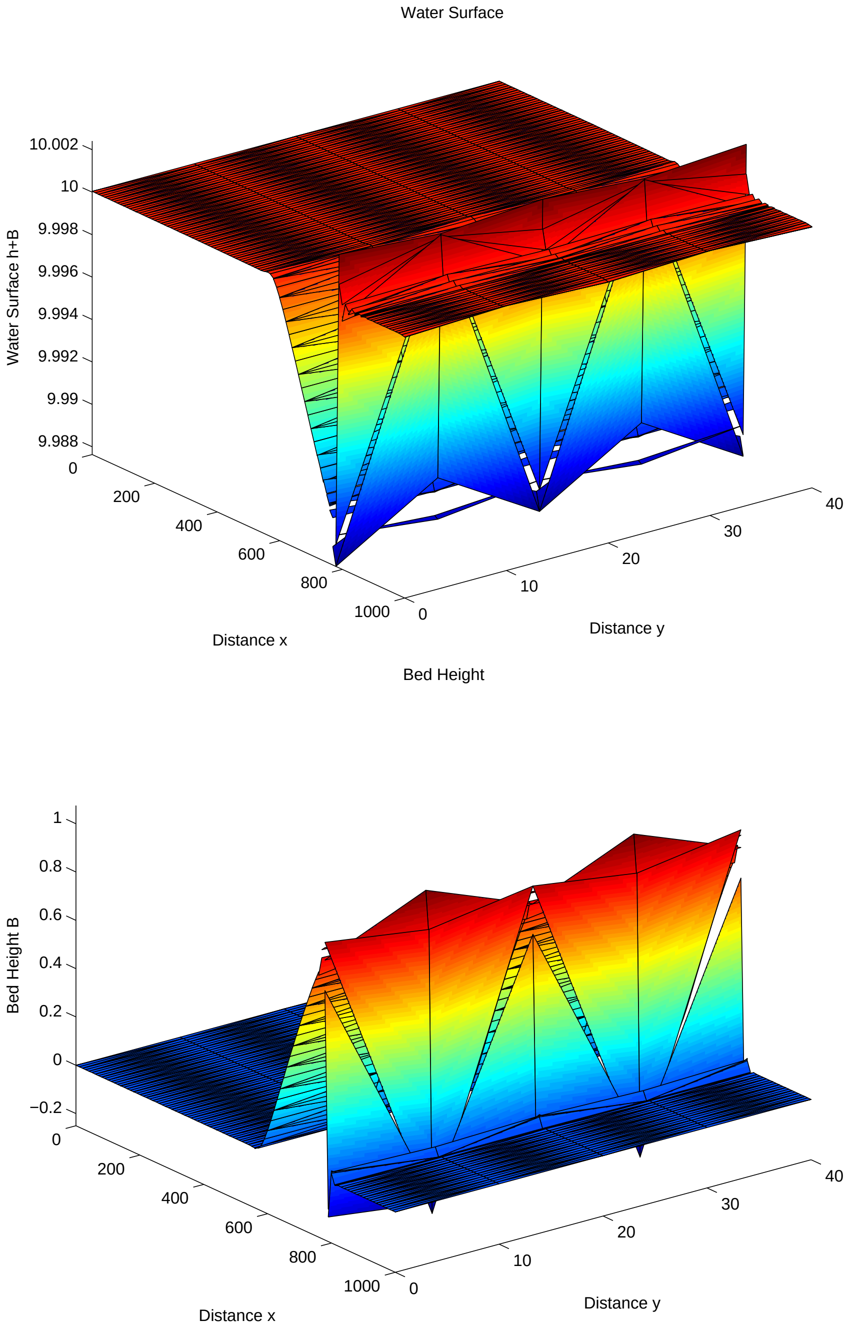


Figure 7.16: Alternate 3D View of Test Case C Results with the Proposed Source Discretisation for Formulation SPLIT-CB Order 2U

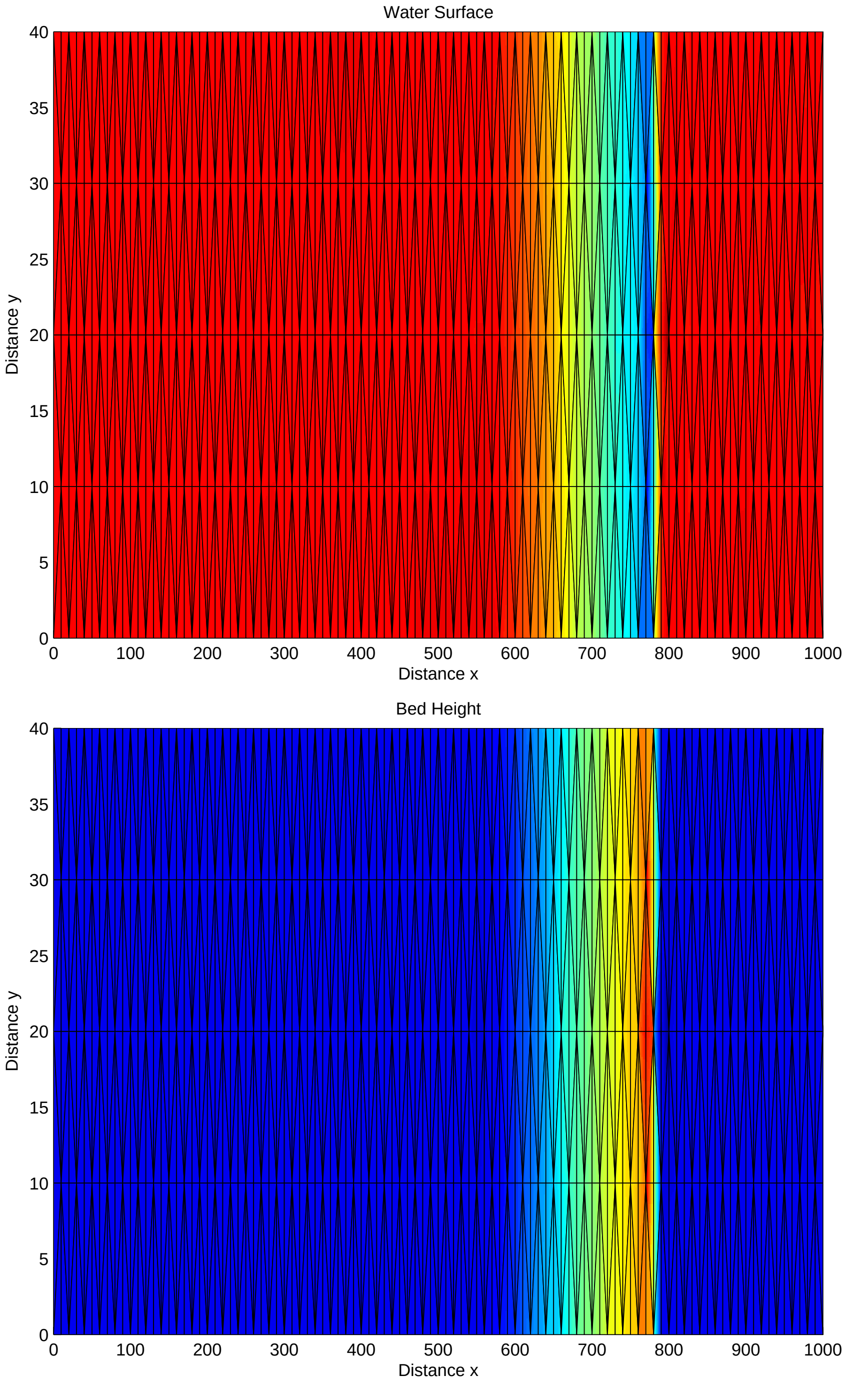


Figure 7.17: 2D Test Case C Results with the Proposed Source Discretisation for Formulation SPLIT-CB Order 2D

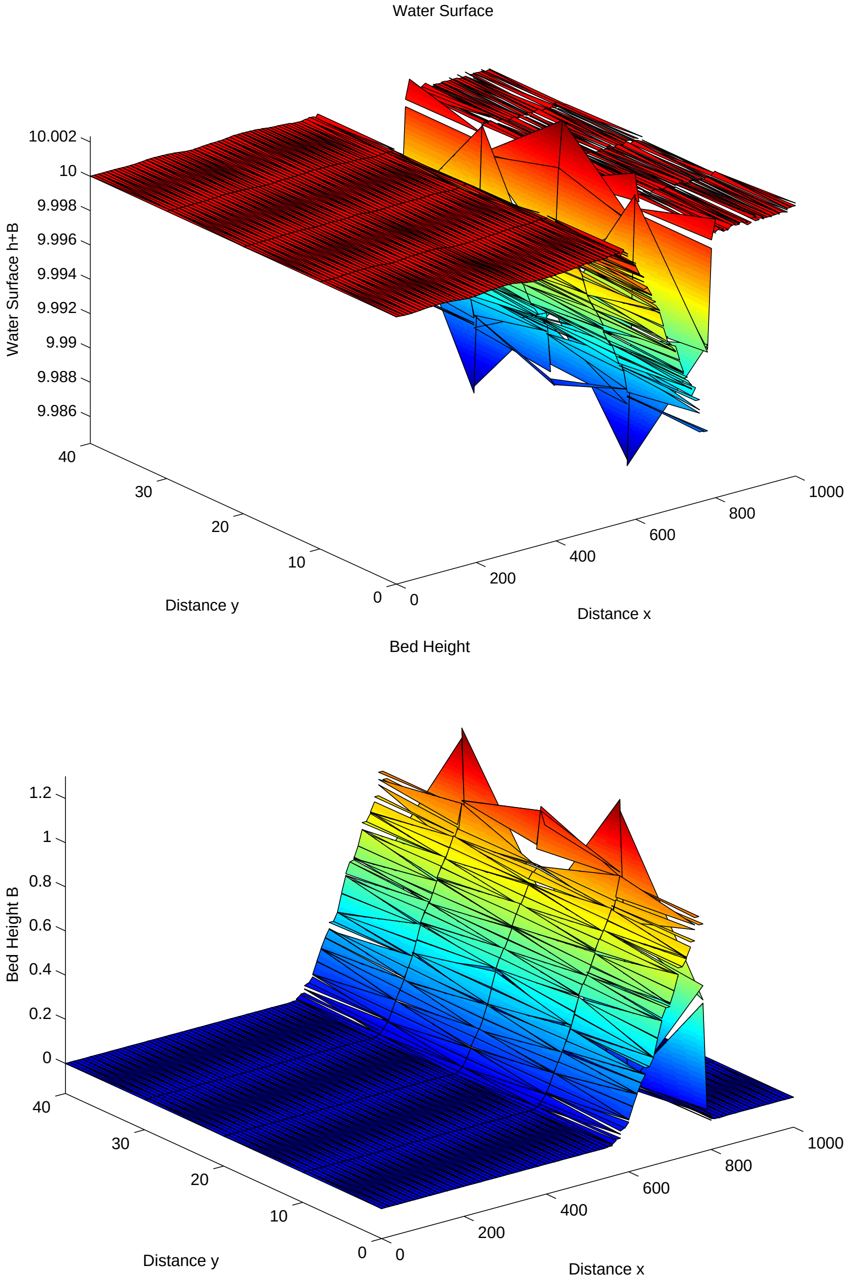
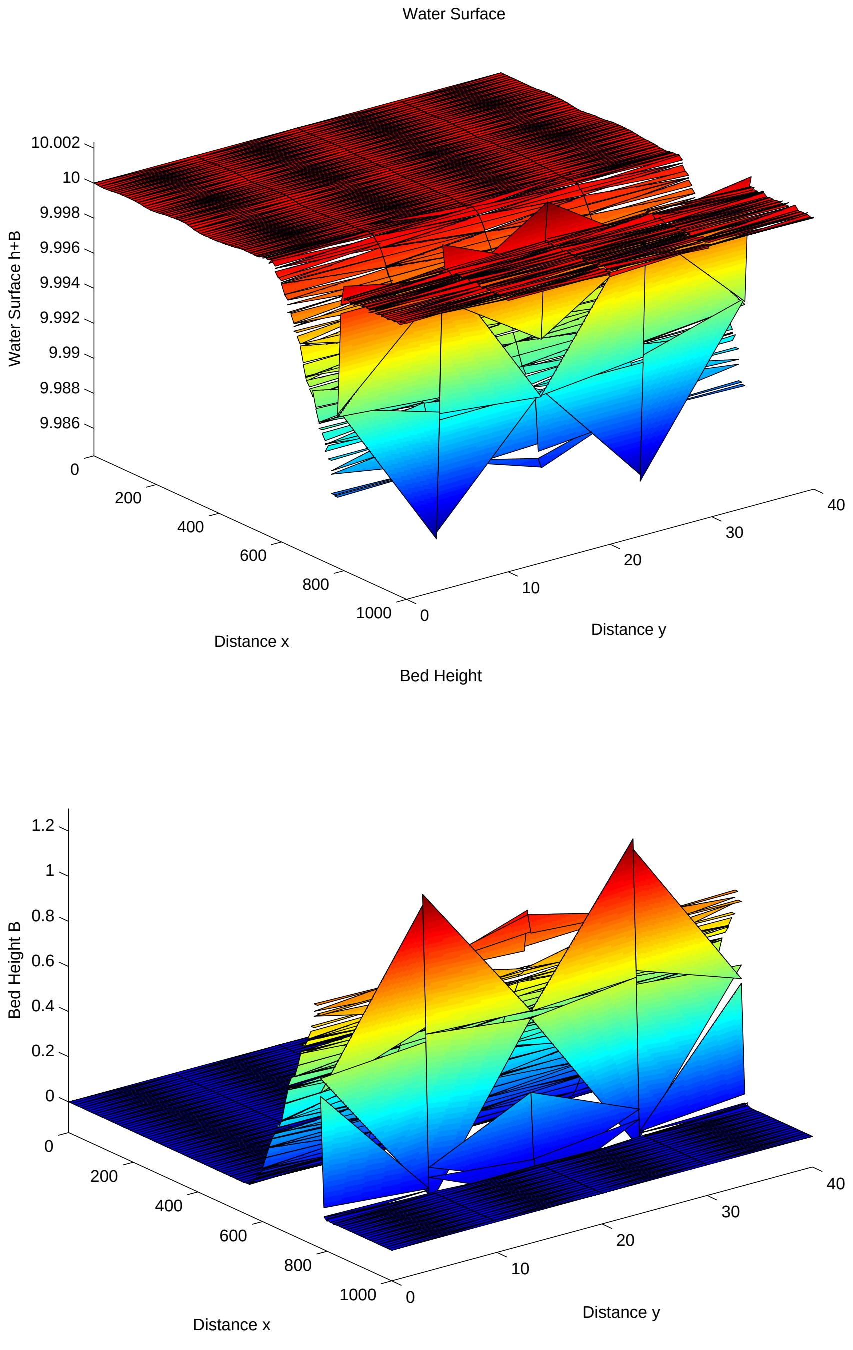


Figure 7.18: 3D Test Case C Results with the Proposed Source Discretisation for Formulation SPLIT-CB Order 2D



Water Surface

Bed Height

Figure 7.19: Alternate 3D View of Test Case C Results with the Proposed Source Discretisation for Formulation SPLIT-CB Order 2D

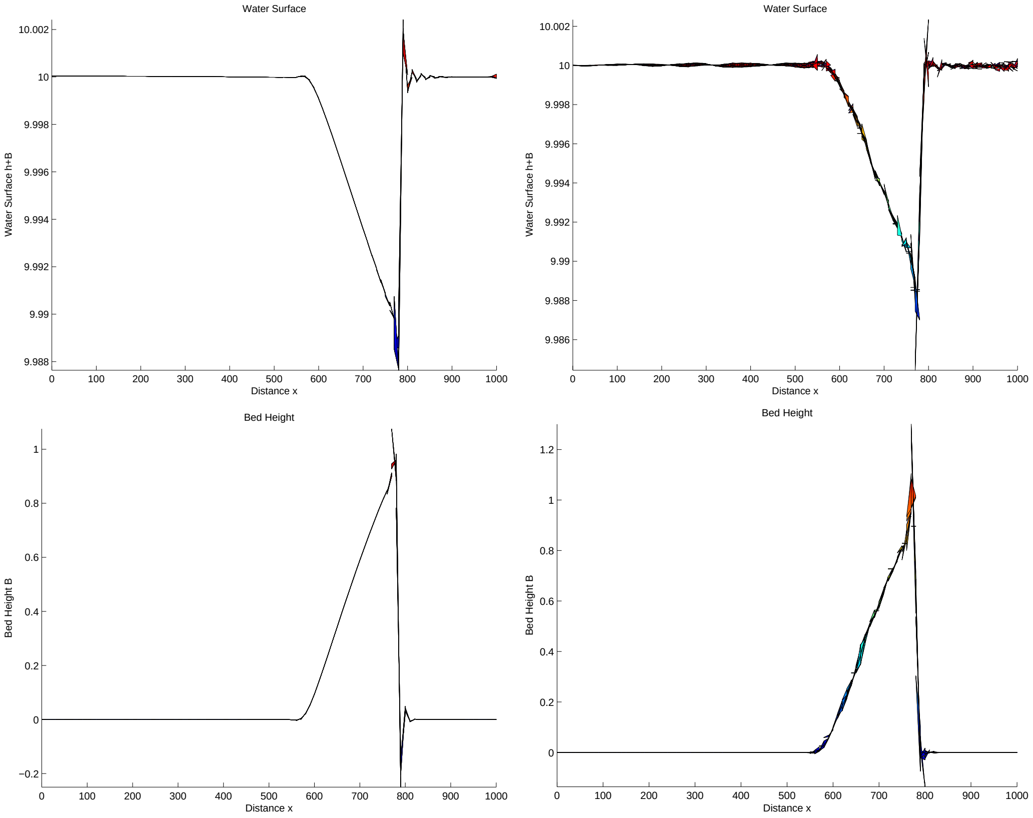


Figure 7.20: Test Case C Results with the Proposed Source Discretisation for Formulation SPLIT-CB Comparison Of Orders

Order 2U is on the left and Order 2D is on the right. The surface is shown at the top and the bed is shown below.

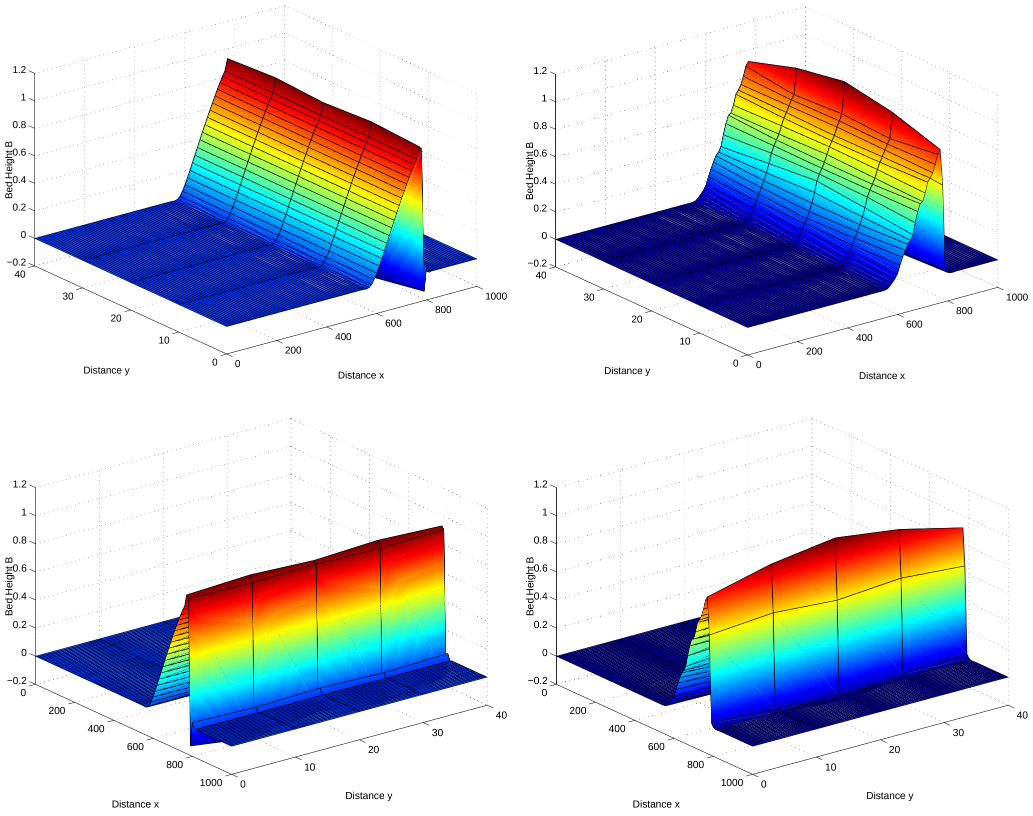


Figure 7.21: Test Case C Mean Results with the Proposed Source Discretisation for Formulation SPLIT-CB Comparison Of Orders

Order 2U is on the left and Order 2D is on the right. Both graphs show the bed from different angles.

7.8 Summary

In this chapter we have seen the progression needed to take the method defined in 1D to 2D. This was achieved through the use of the 2D RKDG method as given by Cockburn *et al.* [18]. We have demonstrated, through test cases, that this method, with the source term discretisation used by Schwanenberg *et al.* [68], fails to satisfy the C-property when the bed is discontinuously represented, as is the case for full morphodynamics.

We have also defined the generalisation of the finite difference source term discretisation used by Hudson [34] to arbitrary triangulations in 2D and combined this with the finite element source term discretisation of Schwanenberg to create a proposed source term discretisation that satisfies the C-property without any assumptions needed on the profile of the bed. This has been demonstrated through proof and test cases.

Chapter 8

Conclusions

8.1 Summary

In this thesis we have seen the development of the RKDG method, from its raw form into a form that is capable of accurately modelling morphodynamics.

We initially described viable formulations of the equations that model hydrodynamics and morphodynamics. These were a subset of the formulations that were modelled by Hudson [34], chosen for their performance in a finite difference setting. We then presented some preliminary description of numerical techniques that can be used and properties that the scheme must satisfy. In particular we presented the C-property that allowed us to say whether the scheme would have the correct steady state solution in the absence of flow.

We have presented the RKDG method for 1D, as given by Cockburn *et al.* [15], in its raw form and the extension for source terms, necessary for hydrodynamics, as given by Schwanenberg [68]. We then demonstrated, through proof and test cases, that the method satisfies the C-property only under the assumption that the bed is continuously represented. We showed that, since this assumption does not apply when the bed is part of the system being modelled, the RKDG method in its raw form does not provide good results for the morphodynamical models.

Following this, we defined a new source term discretisation by combining the source term discretisation given by Schwanenberg and the finite difference source term given by Hudson. We have proved that this source term discretisation satisfies the C-property requirements without the assumption on the bed profile needed for the raw form. Because of this, we have been able to demonstrate, through test cases, the C-property satisfaction of the scheme and the suitability to model morphodynamics.

We also considered the process of splitting the morphodynamics and hydrodynamics with the aim of solving these with as large a time step as stability will allow. This has the benefit of simplifying the equations being solved and increasing the computational efficiency of the scheme. We demonstrated that, although the transport speed was inaccurate, the difference between the full system approach and the splitting approach was minimal.

We also demonstrated the need to consider how the temporal extrapolation of the solution between the morphodynamics and hydrodynamics is achieved in the split approach. We affirmed, in Section 6.1.2, that to get a scheme that is n^{th} order accurate in time and space the water needs to be iterated n times over each bed step, *i.e.*once for each RK time step iteration. We showed that a backwards, in time, extrapolation of the bed profile generated significantly better results than forward extrapolation.

The result of this work is two approaches to modelling morphodynamics using the RKDG method. A full system approach gives a more accurate result at the expense of more computation. The split approach with backward extrapolation gave very good results with a better computational efficiency. In both cases the results were free of spurious oscillations and better than the finite difference approaches of Hudson.

We then extended the knowledge gained from 1D to 2D. We presented the 2D RKDG method, as given by Cockburn *et al.* [18], and its extension to source terms. The new source term discretisation was defined for 2D which required the extension of the finite difference source term of Hudson to arbitrary triangulations. We have proved that this new source term discretisation gives a scheme that satisfies the C-property without any assumptions necessary on the bed profile and provided test cases and results for demonstrating this C-property satisfaction in 2D.

Test cases for modelling morphodynamics in 2D are provided and we suggest, given the information discovered in 1D and 2D, that scheme with the proposed source term discretisation will give good numerical results for morphodynamics that will rival, or even exceed, those of finite difference methods. This, combined with the natural ability of the scheme to use arbitrary triangulations and irregular meshes, means that the scheme is particularly suited to the application of modelling morphodynamics.

8.2 Further Work

Further work in this area must initially begin with the verification of the given method in 2D. It is important to demonstrate that the given extensions to the raw method actually give the results that were inferred at the end of the previous chapter. This is simply a matter of implementing and testing the method on the given test problems. Comparisons can be drawn with the work of Hudson [34] to identify if the improvements in accuracy, that were observed in 1D, extend to 2D. Following this verification the scope for extending the RKDG method is opened to a wide variety of areas.

Since the RKDG method is a finite element method its extension from 1D to 2D was relatively simple. This is also true for the extension to 3D. The next step in the development of the RKDG method for the morphodynamical equations could see the introduction of variations in the solution in the vertical axis.

The RKDG method provides an excellent basis for research into other numerical techniques. The RKDG method naturally supports *h-p* refinement. The naturally local nature of the RKDG method means that the use of a higher order approximation in a cell does not affect the evaluation of any other cells and cells can easily be split or merged without the need to worry about maintaining continuity. In addition, the definition of the new source term naturally suggests its definition for higher orders of accuracy.

In morphodynamics the shape of rivers and estuaries change over long times and this effect can be significant on the time scales that morphodynamics are usually modelled over. It is possible that, in particular regions, deposit is so great that the regions become dry. Equivalently scour can create new or larger flow regions. In industry there is currently extensive research into a natural process of adding and removing cells in the mesh, known as wetting and drying. Since the boundary treatment for the RKDG method is simple, it can naturally be combined with these wetting and drying processes [5, 43]. An alternative approach to wetting and drying is the moving mesh methods which academia is currently showing interest in. Again, these methods can be implemented easily with the RKDG method [47].

8.3 Post Note

After the original submission of this thesis, Xing and Shu published a paper in the Journal of Computational Physics entitled "High Order Well-Balanced Finite Volume WENO Schemes and Discontinuous Galerkin Methods for a Class of Hyperbolic Systems with Source Terms" [79]. In this paper they defined balanced WENO and discontinuous Galerkin schemes that satisfy a source/flux balance requirement. For the shallow water equation this balance requirement is the C-property.

As their work was produced independently of this thesis it has not been referred to within the thesis itself. In this section we shall review the paper and compare it to the work of this thesis.

8.3.1 Overview of the Paper

The paper introduces the approaches needed to modify the finite volume WENO (Weighted Essentially Non-Oscillatory) scheme to achieve a balance between the flux term discretisation and the source term discretisation. This is then used to infer the modification needed for the RKDG scheme to achieve the same balance.

The paper provides, initially, a review of both WENO schemes and RKDG schemes before deriving the discretisations needed to balance the WENO scheme. This is followed by an inference of the discretisation needed for the RKDG scheme. For both, WENO and RKDG, schemes a proof is provided to demonstrate that the balance is achieved.

The paper provides examples of the application of this discretisation method with the SWE in 1D and 2D, the elastic wave equation in 1D and also chemosensitive movement in 1D. It provides numerical evidence through test cases and accuracy tables to demonstrate that the balance has been achieved whilst maintaining high order accuracy.

8.3.2 The Scheme

As this thesis only considers the RKDG method we will only discuss the comparison between the RKDG scheme in the paper and the proposed scheme in this thesis. To achieve C-property satisfaction they write the source term as a cell boundary term, a cell centred term and a high order integral. The effectiveness of the discretisation comes through choosing a suitable discretisation of the high order integral. To demonstrate the principles they use the Lax-Friedrichs numerical flux function and give the source term discretisation needed. Unfortunately, this source term discretisation requires a modification to the Lax-Friedrichs flux function to maintain enough artificial vicosity.

8.3.3 Results

In the paper several results are presented for the 1D SWE. These include two tests that separately demonstrate the results given in Test A of this thesis. In addition they provide a test case from LeVeque, similar in form to Test C in this thesis. They demonstrate good results by comparing the method with a range of grid cell numbers to the same method with 3000 grid cells. They also provide similar test in 2D for which the specification is given by,

$$[x, y, t] \equiv [0, 2] \times [0, 1] \times [0, 0.6],$$

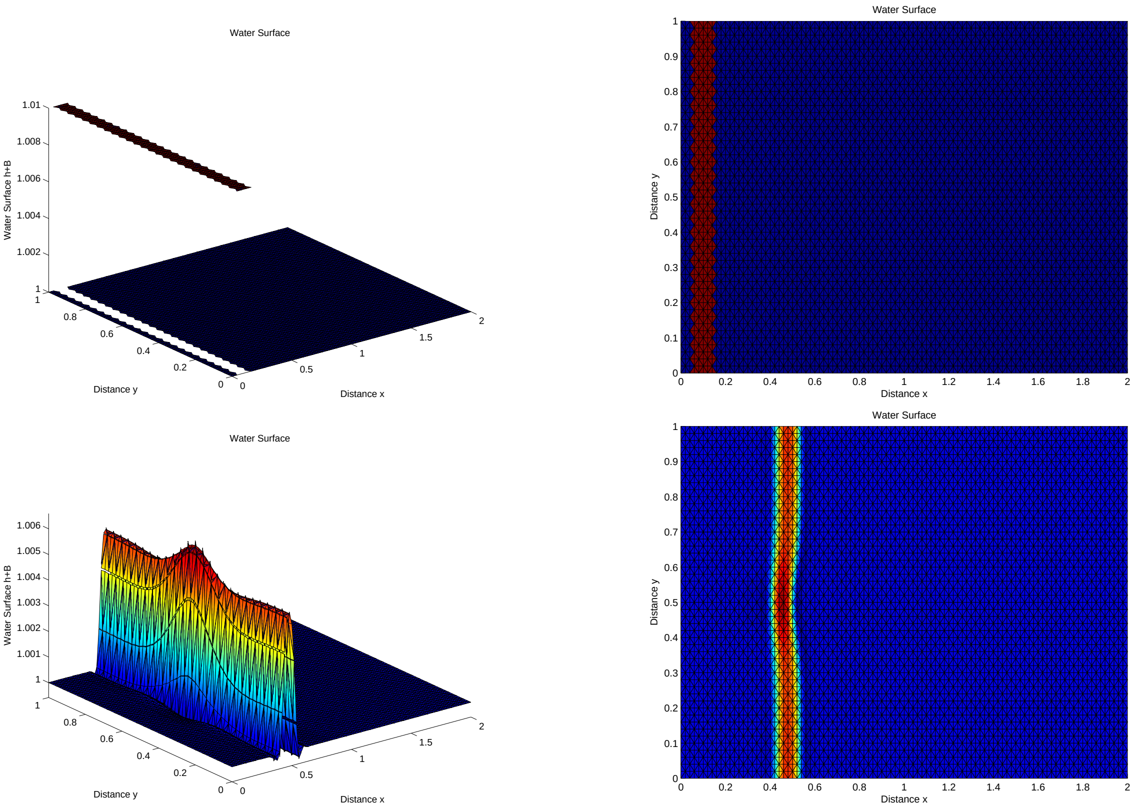


Figure 8.1: Results for Formulation SWE-C Order 2U for 2D Test Case presented in [79] - Part 1.

Top - $t = 0$, bottom $t = 0.12$.

$$B(x, y) = 0.8e^{-5(x-0.9)^2-50(y-0.5)^2},$$

$$h(x, y, 0) = \begin{cases} 1 - b(x, y) + 0.01, & \text{if } 0.05 \leq x \leq 0.15, \\ 1 - b(x, y), & \text{otherwise,} \end{cases}$$

$$u(x, y, 0) = v(x, y, 0) = 0,$$

$$g = 9.812, \qquad \alpha = 0.$$

Demonstration of the results for the this test with the method proposed in this thesis is given in Figure 8.1 to Figure 8.6 with a grid resolution of 100 by 50 cells. These agree with the results obtained in the paper.

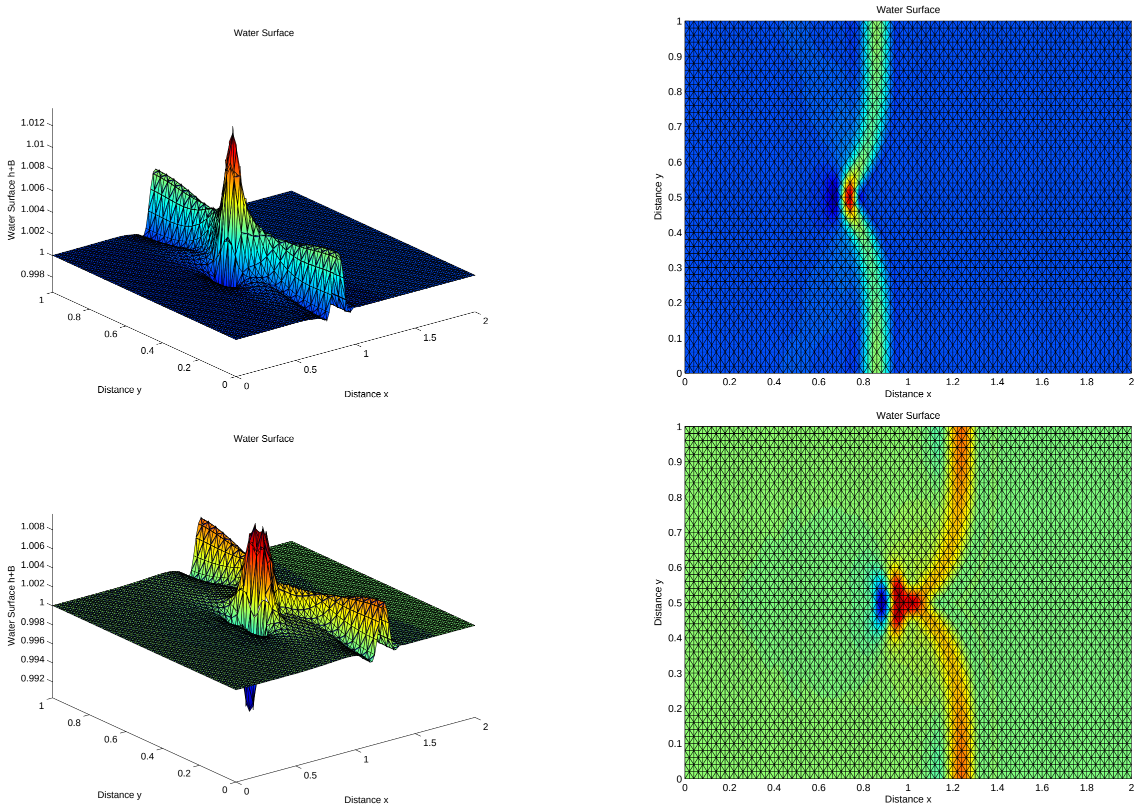


Figure 8.2: Results for Formulation SWE-C Order 2U for 2D Test Case presented in [79] - Part 2.

Top - $t = 0.24$, bottom $t = 0.36$.

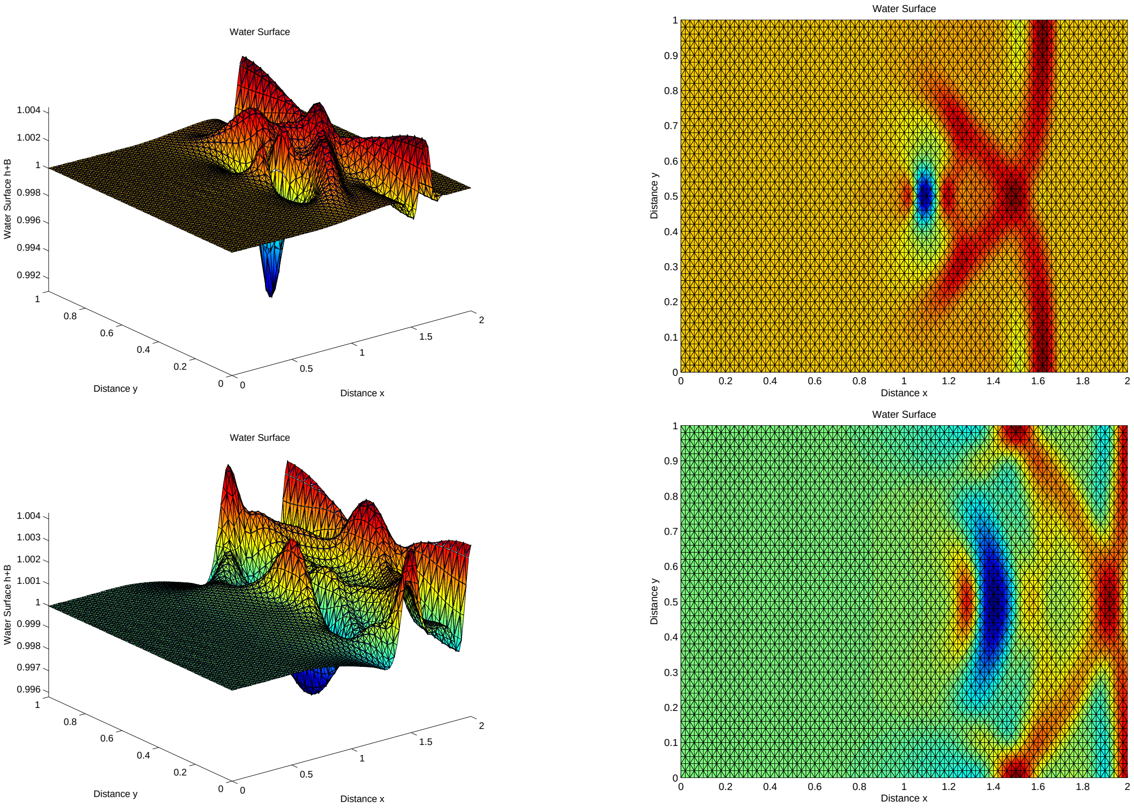


Figure 8.3: Results for Formulation SWE-C Order 2U for 2D Test Case presented in [79] - Part 3.

Top - $t = 0.48$, bottom $t = 0.60$.

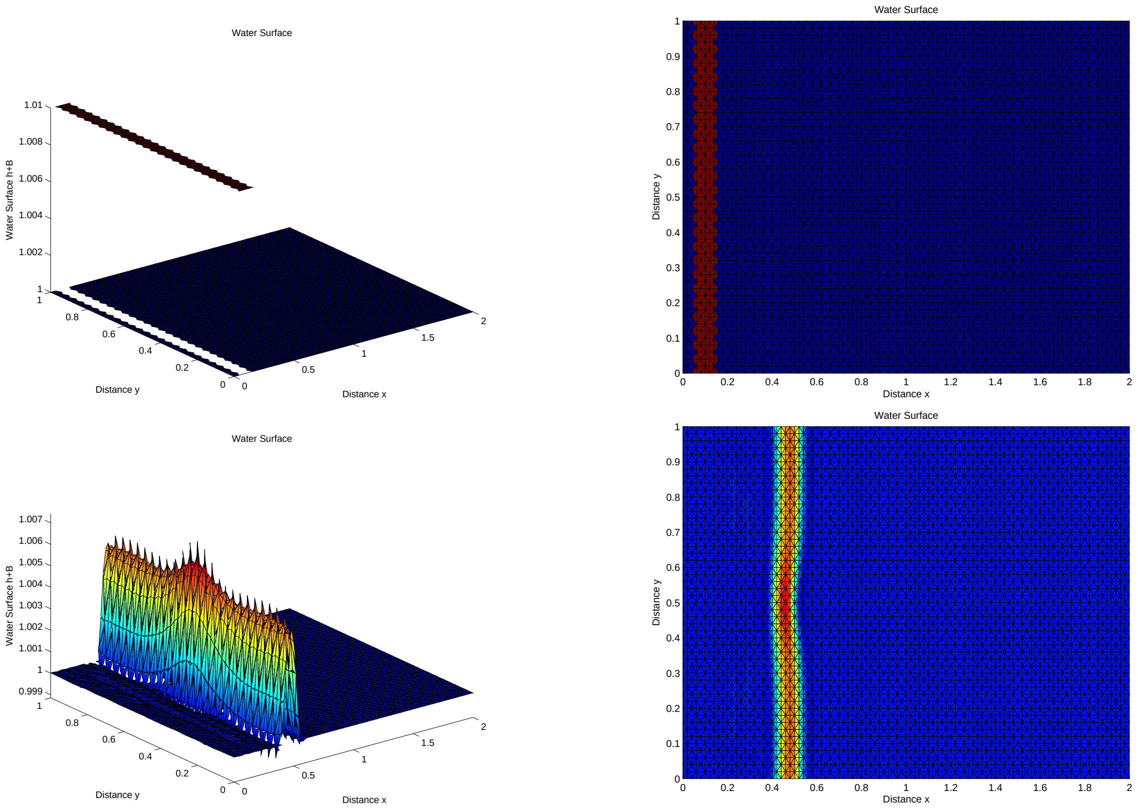


Figure 8.4: Results for Formulation SWE-C Order 2D for 2D Test Case presented in [79] - Part 1.

Top - $t = 0$, bottom $t = 0.12$.

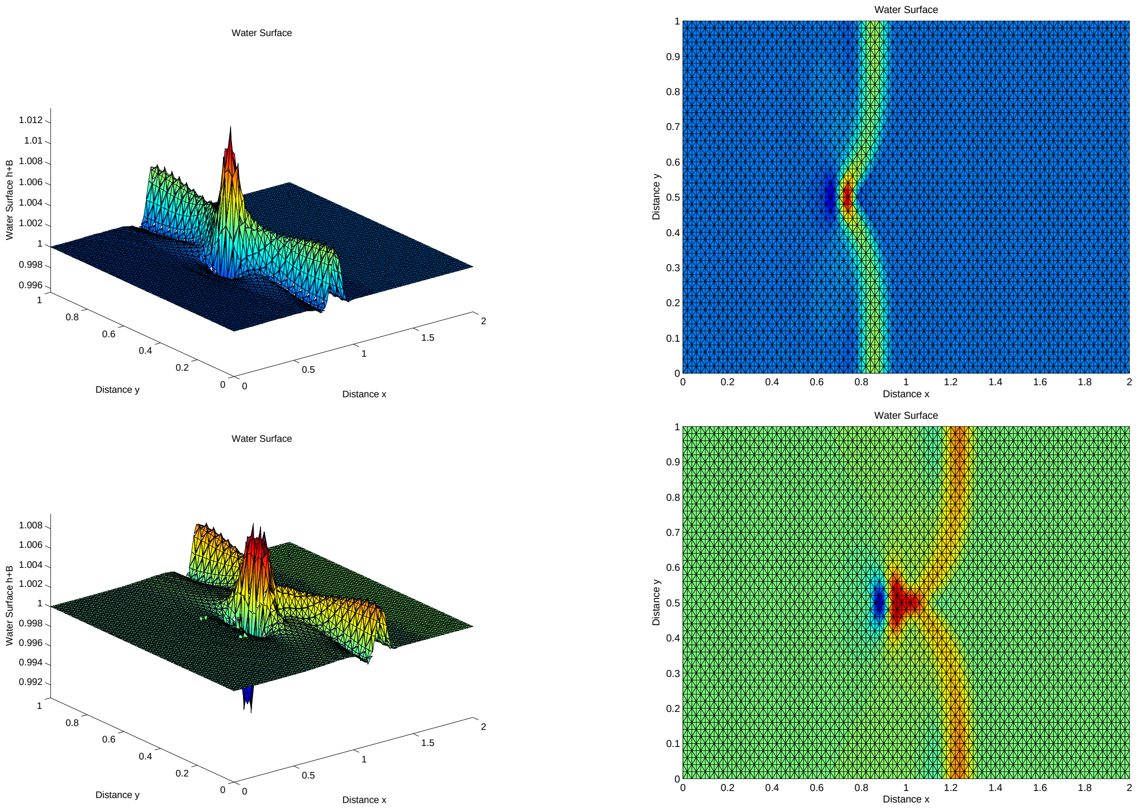


Figure 8.5: Results for Formulation SWE-C Order 2D for 2D Test Case presented in [79] - Part 2.

Top - $t = 0.24$, bottom $t = 0.36$.

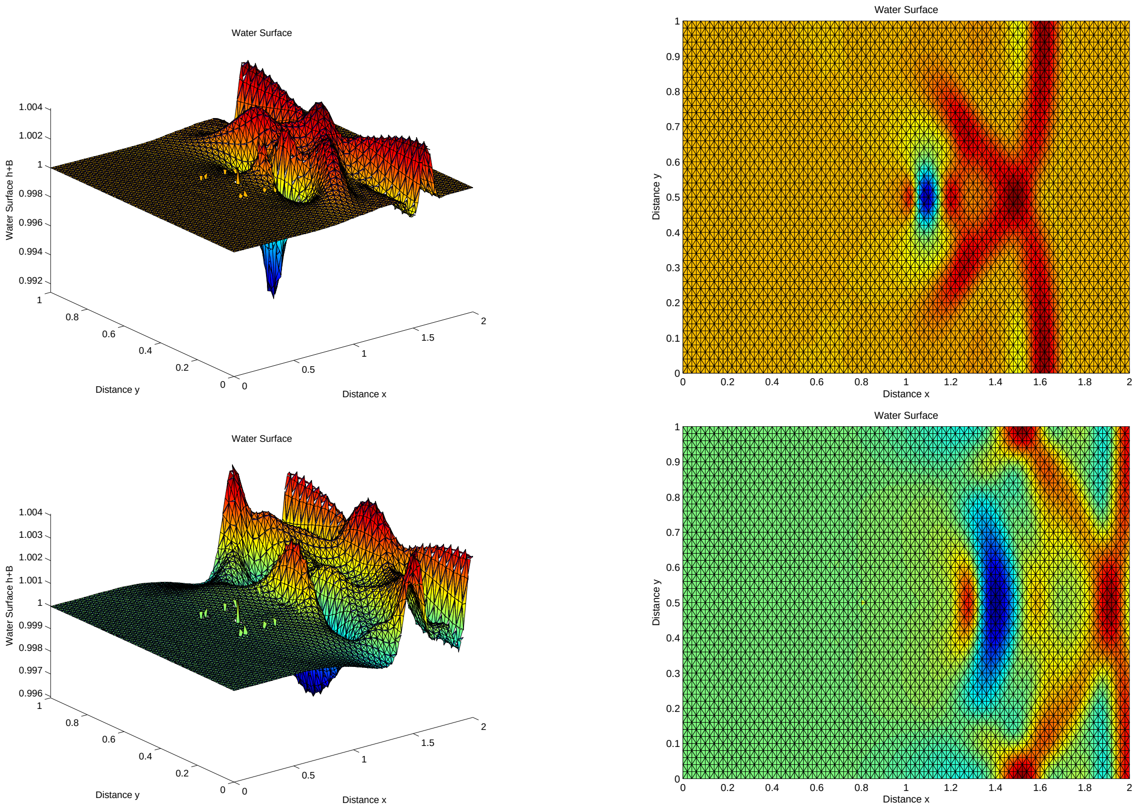


Figure 8.6: Results for Formulation SWE-C Order 2D for 2D Test Case presented in [79] - Part 3.

Top - $t = 0.48$, bottom $t = 0.60$.

8.3.4 A Comparison of Methods

It is clear that both their proposed method and the one formulated in this thesis require the need to generate a balancing source term discretisation in a finite difference setting which then forms the basis of the discretisation for the RKDG method. Their proposed discretisation benefits from a more rigorous derivation, and it is thus easier to prove properties of the scheme, however the discretisation in this thesis is simple to manipulate and requires no modification of the numerical flux function to achieve the balance. Table 6.1, in comparison to similar tables in [79], show that both schemes are high-order accurate, satisfy the C-property and generate good results for the test case shown here.

8.3.5 Finally...

It should be emphasised that the paper described in this section had not been published by the time that this thesis was originally submitted and was not referred to during the course of this Ph.D.

Bibliography

[1] V. Aizinger, C. Dawson, B. Cockburn, and P. Castillo. The local discontinuous Galerkin method for contaminant transport. *Adv. Water Resources*, 24:73–87, 2001.

[2] V. R. Ambati and O. Bokhove. A space-time discontinuous Galerkin finite element method for shallow water flows. Submitted for publication, 2006.

[3] K. E. Atkinson. *Numerical Analysis*. John Wiley & Sons, 1989.

[4] P. D. Bates and M. G. Anderson. A two-dimensional finite element method for river flow inundation. In *Proceedings: Mathematical and Physical Sciences*, volume 440, pages 481–491, 1993.

[5] P. D. Bates and J.-M. Hervouet. A new method for moving-boundary hydrodynamic problems in shallow water. In *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, volume 455, pages 3107–3218, 1999.

[6] P. Batten, C. Lambert, and D. M. Causon. Positively conservative high-resolution convection schemes for unstructured elements. *Internat. J. Numer. Methods Engrg.*, 39:1821–1838, 1996.

[7] A. Bermudez and M. E. Vasquez. Upwind methods for hyperbolic conservation laws with source terms. *Comput. & Fluids*, 23:1049–1071, 1994.

[8] A. Burbeau, P. Sagaut, and Ch.-H. Bruneau. A problem-independent limiter for higher-order Runge-Kutta discontinuous Galerkin methods. *J. Comput. Phys.*, 169:111–150, 2001.

[9] J. Burguete and P. Garcia-Navarro. Efficient construction of high-resolution TVD conservative schemes for equations with source terms. application to shallow water flows. *Int. J. Num. Meth. Fluids*, 37(2):209–248, 2001.

[10] G. Chavant and B. Cockburn. The local projection P^0P^1 discontinuous Galerkin finite element method for scalar conservation laws. M^2AN, 23:565–592, 1989.

[11] G. Chavant and G. Salzano. A finite element method for the 1d water flooding problem with gravity. *J. Comput. Phys.*, 45:307–344, 1982.

[12] B. Cockburn. Discontinuous Galerkin methods for convection dominated problems. Lecture notes.

[13] B. Cockburn, S. Hou, and C.-W. Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.*, 54:545–581, Apr 1990.

[14] B. Cockburn, G. E. Karniadakis, and C.-W. Shu. *The Development of Discontinuous Galerkin Methods*, pages 3–50. Springer, 2000.

[15] B. Cockburn, S. Y. Lin, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *J. Comput. Phys.*, 84:90–113, 1989.

[16] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Math. Comp.*, 52:411–435, Apr 1989.

[17] B. Cockburn and C.-W. Shu. The Runge-Kutta local projection P^1 discontinuous Galerkin method for scalar conservation laws. M^2AN, 25:337–351, 1991.

[18] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *J. Comput. Phys.*, 141:199–224, 1998.

[19] G. Cohen, X. Ferrieres, and S. Pernet. A spatial high-order hexahedral discontinuous Galerkin method to solve Maxwell's equations in time domain. *J. Comput. Phys.*, 217:340–363, 2006.

[20] N. Crnjaric-Zic, S. Vukovic, and L. Sopta. Extension of ENO and WENO schemes to one-dimensional sediment transport equations. *Comput. & Fluids*, 33:31–56, 2004.

[21] C. Dawson, J. Westerink, J. Feyen, and D. Pothina. Continuous, discontinuous and coupled discontinuous-continuous Galerkin finite element methods for the shallow water equations. *Int. J. Num. Meth. Fluids*, 52:63–88, 2006.

[22] C. Eskilsson and S. J. Sherwin. A triangular spectral/hp discontinuous Galerkin method for modelling 2D shallow water equations. *Int. J. Num. Meth. Fluids*, 45:605–623, 2004.

[23] C. Eskilsson and S. J. Sherwin. Spectral/*hp* discontinuous Galerkin methods for modelling 2D Boussinesq equations. *J. Comput. Phys.*, 212:566–589, 2006.

[24] T. Gallouet, J.-M. Herard, and N. Seguin. Some approximate Godunov schemes to compute shallow-water equations with topography. *Comput. & Fluids*, 32:479–513, 2003.

[25] F. Giraldo, J. Hesthaven, and T. Warburton. Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *J. Comput. Phys.*, 181:499–525, 2002.

[26] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. Technical Report 96-50, ICASE, 1996.

[27] S. Gottlieb and C.-W. Shu. Total-variation-diminishing Runge-Kutta schemes. *Math. Comp.*, 67:73–85, 1998.

[28] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability preserving high-order time discretization methods. Technical Report 2000-15, ICASE, 2000.

[29] A. J. Grass. Sediment transport by waves and currents. Technical Report FL29, SERC London Centre for Marine Technologies, 1981.

[30] R. J. Hardy, P. D. Bates, M. G. Anderson, C. Moulin, and J.-M. Hervouet. Development of a reach scale two-dimensional finite element model for floodplain sediment deposition. *Proceedings of the Institution of Civil Engineers. Water, Maritime and Energy*, 142:141–156, 2000.

[31] M. Horritt. Development and testing of a simple 2d finite volume model of sub-critical shallow water flow. *Int. J. Num. Meth. Fluids*, 44:1231–1255, 2004.

[32] M. E. Hubbard. Multidimensional slope limiters for MUSCL-type finite volume schemes. *J. Comput. Phys.*, 155:54–74, 1998.

[33] M. E. Hubbard and P. Garcia-Navarro. Flux difference splitting and the balancing of source terms and flux gradients. *J. Comput. Phys.*, 165:89–125, 2000.

[34] J. Hudson. *Numerical Techniques For Morphodynamic Modelling.* PhD thesis, University Of Reading, Department Of Mathematics, University Of Reading, Whiteknights, Reading, RG6 6AX, October 2001.

[35] G. B. Jacobs and J. S. Hesthaven. High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids. *J. Comput. Phys.*, 214:96–121, 2006.

[36] C. Johnson, U. Navert, and J. Pitkaranta. Finite element methods for linear hyperbolic problems. *Comput. Methods Appl. Mech. Engrg.*, 45:285–312, 1984.

[37] S. Karni. Far field behaviour and far field boundary conditions - a numerical study. Technical Report 8711, Cranfield Institute Of Technology, College Of Aeronautics, 1987.

[38] S. Karni. The problem of far field boundaries - slowing down the outgoing waves. Technical Report 8721, Cranfield Institute Of Technology, College Of Aeronautics, 1987.

[39] S. Karni. One way absorbing far field boundaries by gradual wave attenuation. Technical Report 8816, Cranfield Institute Of Technology, College Of Aeronautics, 1988.

[40] T. Katsaounis and C. Simeoni. Second order approximation of the viscous Saint-Venant system and comparison with experiments. In *Hyperbolic Problems: Theory, Numerics and Applications*. Springer Verlag, 2003.

[41] R. L. Kolar, K. M. Dresback, C. M. Szpilka, J. H. Atkinson, E. M. Tromble, T. C. G. Kibbey, R. A. Richard, and J. L. Hoggan. A comparison of continuous and discontinuous galerkin algorithms for shallow water transport. In *Proceedings: 9th International Conference on Estuarine and Coastal Modelling*, 2005.

[42] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, and J. E. Flaherty. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *J. Appl. Num. Math.*, 48(3):323–338, 2004.

[43] S. Kruger and P. Rutschmann. Modelling 3d supercritical flow with extended shallow-water approach. *J. Hydr. Engrg.*, 132:916–926, 2006.

[44] E. J. Kubatko, J. J. Westerink, and C. Dawson. An unstructured grid morphodynamic model with a discontinuous Galerkin method for bed elevation. *Ocean Modelling*, 15:71–89, 2006.

[45] R. J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhauser, 1992.

[46] R. J. LeVeque. Balancing source terms and flux gradients in high-resolution Godunov methods: The quasi-steady wave propogation algorithm. *J. Comput. Phys.*, 146, 1998.

[47] R. Li and T. Tang. Moving mesh discontinuous galerkin method for hyperbolic conservation laws. *J. Sci. Comp.*, 27:347–363, 2006.

[48] G.-F. Lin, J.-S. Lai, and W.-D. Guo. Finite-volume component-wise TVD schemes for 2D shallow water equations. *Advances in Water Resources*, 26:861–873, 2003.

[49] H. Liu and J. Yan. A local discontinuous Galerkin method for the Korteweg-de Vries equation with boundary effect. *J. Comput. Phys.*, 215:197–218, 2006.

[50] R. B. Lowrie, P. L. Roe, and B. van Leer. A space-time discontinuous Galerkin method for the time-accurate numerical solution of hyperbolic conservation laws. In *AIAA Computational Fluid Dynamics Conference*, pages 135–150, 1995.

[51] M. Lukacova-Medvidova and Z. Vlk. Well-balanced finite volume evolution Galerkin methods for the shallow water equations with source terms. *Int. J. Num. Meth. Fluids*, 47:1165–1171, 2005.

[52] H. Luo, J. D. Baum, and R. Lohner. A *p*-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids. *J. Comput. Phys.*, 211:767–783, 2006.

[53] E. Marchandise, J.-F. Remacle, and N. Chevaugeon. A quadrature free discontinuous Galerkin method for the level set equation. *J. Comput. Phys.*, 212:338–357, 2006.

[54] C. R. Nastase and D. J. Mavriplis. High-order discontinuous Galerkin methods using an *hp*-multigrid approach. *J. Comput. Phys.*, 213:330–357, 2006.

[55] D.J. Needham and R. D. Hey. On nonlinear simple waves in alluvial river flows: A theory for sediment bores. *Phil. Trans. R. Soc. Lond.*, 334:25–53, 1991.

[56] S. Noolle, N. Pankratz, G. Puppo, and J. R. Natvig. Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. *J. Comput. Phys.*, 213:474–499, 2006.

[57] W. Ottevanger. Discontinuous finite element modelling of river hydraulics and morphology. Master's thesis, University of Twente, Netherlands, 2005.

[58] E. Pichelin and T. Coupez. A Taylor discontinuous Galerkin method for the thermal solution in 3D mold filling. *Comput. Methods Appl. Mech. Engrg.*, 178:153–169, 1998.

[59] A. Priestly. The Taylor-Galerkin method for the shallow-water equations on the sphere. *Monthly Weather Review*, 120:3003–3015, 1992.

[60] J. Qiu, M. Dumbser, and C.-W. Shu. The discontinuous Galerkin method with Lax-Wendroff type time discretizations. *Comput. Methods Appl. Mech. Engrg.*, 194:4528–4543, 2005.

[61] J. Qiu, B. C. Khoo, and C.-W. Shu. A numerical study for the performance of Runge-Kutta discontinuous Galerkin method based on different numerical fluxes. *J. Comput. Phys.*, 212:540–565, 2006.

[62] J. Qiu and C.-W. Shu. Runge-Kutta discontinuous Galerkin method using WENO limiters. *J. Sci. Comp.*, 26:907–929, 2005.

[63] P. Rasetarinera and M. Y. Hussaini. An efficient implicit discontinuous spectral Galerkin method. *J. Comput. Phys.*, 172:718–738, 2001.

[64] W. H. Read and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[65] J.-F. Remacle, S. S. Frazao, X. Li, and M. S. Shephard. Adaptive discontinuous Galerkin method for the shallow water equations. *Int. J. Numer. Meth. Fluids*, 52:903–923, 2006.

[66] M. Restelli, L. Banavenhira, and R. Sacco. A semi-Lagrangian discontinuous Galerkin method for scalar advection by incompressible flows. *J. Comput. Phys.*, 216:195–215, 2006.

[67] W. J. Rider and R. B. Lowrie. The use of classical Lax-Friedrichs Riemann solvers with discontinuous galerkin methods. *Int. J. Numer. Meth. Fluids*, 40(1-4):479–486, 2002.

[68] D. Schwanenberg and J. Kongeter. *A Discontinuous Galerkin Method for the Shallow Water Equations with Source Terms*, pages 419–424. Springer, 2000.

[69] C.-W. Shu. Total-variation-diminishing time discretizations. *J. Sci. Stat. Comput.*, 9:1073–1084, 1988.

[70] R. L. Soulsby. *Dynamics of Marine Sands: A Manual for Practical Applications.* Thomas Telford, 1997.

[71] R. L. Soulsby. Dynamics of marine sands, a manual for practical applications. Technical Report Report SR 466, H. R. Wallingford, 1997.

[72] M. R. Spiegel and J. Liu. *Mathematical Handbook of Formulas and Tables.* McGrawHill, second edition, 1999.

[73] E. F. Toro. *Shock-Capturing Methods for Free-Surface Shallow Flows.* Wiley, 2001.

[74] S. Tu and S. Alibadi. A slope limiting proceedure in discontinuous Galerkin finite element method for gas dynamics applications. *Int. J. Numer. Anal. Model.*, 2(2):163–178, 2005.

[75] J. J. W. van der Vegt. Space-time discontinuous Galerkin finite element method with dynamics grid motion for inviscid compressible flows: 1. general formulation. *J. Comput. Phys.*, 182:546–585, 2002.

[76] L. C. Van-Rign. *Sediment Transport in Rivers, Estuaries and Coastal Seas.* Aqua Publications, 1993.

[77] G. Whitham. *Linear and Nonlinear Waves.* Wiley-Interscience, 1974.

[78] W. Wu, W. Rodi, and T. Wenka. 3D numerical modeling of flow and sediment transport in open channels. *J. Hydr. Engrg.*, 126:4–15, 2000.

[79] Y. Xing and C.-W. Shu. High order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. *J. Comput. Phys*, 214:576–598, 2006.