


5-2012

Mathematical Modeling of Fluid Spills in Hydraulically Fractured Well Sites

Oluwafemi Michael Taiwo
University of Arkansas, Fayetteville

Follow this and additional works at: <http://scholarworks.uark.edu/etd>

 Part of the [Applied Mathematics Commons](#), [Environmental Engineering Commons](#), [Hydraulic Engineering Commons](#), [Natural Resources and Conservation Commons](#), [Natural Resources Management and Policy Commons](#), and the [Oil, Gas, and Energy Commons](#)

Recommended Citation

Taiwo, Oluwafemi Michael, "Mathematical Modeling of Fluid Spills in Hydraulically Fractured Well Sites" (2012). *Theses and Dissertations*. 328.
<http://scholarworks.uark.edu/etd/328>

This Dissertation is brought to you for free and open access by ScholarWorks@UARK. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of ScholarWorks@UARK. For more information, please contact scholar@uark.edu, ccmiddle@uark.edu.

**MATHEMATICAL MODELING OF FLUID SPILLS IN
HYDRAULICALLY FRACTURED WELL SITES**

**MATHEMATICAL MODELING OF FLUID SPILLS IN HYDRAULICALLY FRACTURED WELL
SITES**

**A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Chemical Engineering**

By

**Oluwafemi Michael Taiwo
Obafemi Awolowo University
Bachelor of Science in Chemical Engineering, 2006**

**May 2012
University of Arkansas**

ABSTRACT

Improved drilling technology and favorable energy prices have contributed to the rapid pace at which the exploitation of unconventional natural gas is taking place across the United States. As a natural gas well is being drilled, reserve pits are constructed to hold the drilling fluids and other materials returned from the drilling process. These reserve pits can fail, and when they do, plant and animal life of the surrounding area may be adversely affected. This project develops a screening tool for a suitable location for a reserve pit. This work will be a critical piece of the Infrastructure Placement Analysis System (IPAS) created by the Low Impact Natural Gas and Oil (LINGO) project.

A terrestrial spill model was developed that can be used as a decision support system in the Fayetteville Shale Play. There is currently no hydraulic model built with oil and gas production sites in mind. The model was developed by using equations describing shallow water flow and the transport of pollutants and sediments. Mass and momentum conservation laws together with physically reasonable assumptions were used in deriving these equations. The equations were solved numerically using the MacCormack time-splitting finite difference scheme. Novel techniques allowed us to solve the shallow water equations over very steep slopes (>30%) and starting with very low fluid depths without encountering unrealistically high values of velocities or negative flow depths.

The model takes as input the following: the Digital Elevation Model (DEM) of the terrain, predominant soil type, amount or rate of spill and initial concentration of pollutant species. The model gives as output the following: flow depth, flow velocities, the areal extent of pollutant contamination and the amount of sediment run-off. Data from the Fayetteville

Shale Play was used in testing the model and the results were as expected. The computer program implementing the model is written in MATLAB. If the program is translated into a more optimally efficient language and run on a supercomputer, it would be computationally fast and still accurate enough to make its use in a real-time decision support system justified.

This dissertation is approved for recommendation
to the Graduate Council.

Dissertation Director:

Dr. Gregory J. Thoma

Dissertation Committee:

Dr. Mark E. Arnold

Dr. Robert R. Beitle, Jr

Dr. Jackson D. Cothren

Dr. Jerry A. Havens

DISSERTATION DUPLICATION RELEASE

I hereby authorize the University of Arkansas Libraries to duplicate this dissertation when needed for research and/or scholarship.

Agreed

Oluwafemi Michael Taiwo

Refused

Oluwafemi Michael Taiwo

ACKNOWLEDGEMENTS

I did not get here by myself. No one gets a PhD or writes a doctoral dissertation in two months without plenty of help. I have benefited greatly from the wisdom of my advisor, Dr. Gregory J. Thoma. He was available at crucial junctures to give much needed direction. This dissertation bears his imprints all around. I am also grateful to the members of my dissertation committee for their guidance and support.

I appreciate the efforts of my friend, Denise Grindstaff, for proofreading this work and making sure it makes sense – to the extent possible – to people of varied technical backgrounds. Thanks to Toyin Aseperi for helping with some of the Appendices, figures, tables and general formatting. I am grateful to the Department of Energy for funding this work.

Lastly, I am grateful to my wife – Abimbola “Aya’ba” Michael-Taiwo – for her constant encouragement.

DEDICATION

To the memory of my mum

Ruth Titilayo Taiwo

(03/29/1952 – 03/12/2002)

TABLE OF CONTENTS

Abstract

Acknowledgements

Dedication

Table of Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Chapter 1	Introduction	
	1.1	World Energy Outlook 1
	1.2	Natural Gas 2
	1.3	Conventional vs. Unconventional Gas 2
	1.4	Hydraulic Fracturing 3
	1.5	Reserve Pit Failures 4
		1.5.1 What is a Reserve Pit? 4
		1.5.2 Contents of a Reserve Pit 5
		1.5.3 Failure of Reserve Pits 7
	1.6	Fayetteville Shale Play 8
	1.7	Aim 11
	1.8	Objectives of This Project 12
Chapter 2	Literature Review	
	2.1	Introduction 13
	2.2	The Shallow Water Equations (SWEs) 13

2.3	Solving the SWEs	16
2.4	Kinematic Wave Model	16
2.5	Diffusion Wave Model	18
2.6	Hydraulic Resistance Parameters	19
2.7	Infiltration Model	24
2.8	Pollutant Transport Model	27
2.9	Erosion and Sediment Transport Models	28
2.10	Numerical Models for Solving the Shallow Water Equations	33
2.11	Finite Element Method	33
2.12	Method of Characteristics (MOC)	34
2.13	Finite Difference Method	36
2.14	Fractional Step Methods	38

Chapter 3 Methodology

3.1	Discretizing the Shallow Water Equations	40
3.2	The MacCormack Method	43
3.3	The Time-Splitting Technique	43
3.4	The Numerical Grid	44
3.5	The MacCormack Time-Splitting Scheme	45
3.6	Courant Condition	50
3.7	Initial and Boundary conditions for the Numerical Scheme	51
	3.7.1 Initial conditions	52
	3.7.2 Boundary conditions	52
3.8	Artificial Viscosity	55
3.9	Negative Flow Depths	58
3.10	Bed Friction Slope	59
3.11	Infiltration Model	62
3.12	Pollutant Transport Model	64
3.13	Erosion-Sediment Transport Module	65
	3.13.1 Sediment Continuity Equation	66
	3.13.2 Calculation Procedure	67

Chapter 4 Results and Discussions

4.1	Introduction	72
4.2	Test Problem 1: A 1D Steady State Problem with no shocks or Discontinuities	72
4.3	Numerical Solution to Test Problem 1	76
4.4	Test Problem 2: A 1D Steady State Problem with Discontinuities	79
	4.4.1 Introduction	79
	4.4.2 Problem Description and its Analytical Solution	79
	4.4.3 Numerical Solution to Test Problem 2	83

4.4.4	Artificial Viscosity applied to the Numerical Solution to Test Problem 2	85
4.4.4.1	Choice of κ when using Artificial Viscosity in A Simulation	87
4.5	Discussion of Model Performance based on the two Test Problems	88
4.6	Introduction to 2D Problems	89
4.7	Influence of Boundary Conditions	89
4.8	Sub-Critical Flow Problem	92
4.9	Flow Over Infiltrating Surfaces	97
4.10	Sediment Delivery	100
4.11	Flow in a Real Terrain with Steep Gradients	101
4.12	Pollutant Transport over real Topography	108
4.13	Sediment Delivery over real Topography	110
Chapter 5	Conclusions and Recommendations	
5.1	Conclusions	113
5.2	Limitations of the Model	115
5.3	Recommendations	116
5.4	Areas of Further Research	117
References		119
Appendices		133

LIST OF FIGURES

Figure 1.1	A drilling site where hydraulic fracturing is taking place	8
Figure 1.2	Producing wells in Arkansas by location	10
Figure 2.7	A Water-Content vs. Depth curve	24
Figure 3.1	Explanation of Unit Width discharges	41
Figure 3.4 (a)	Typical staggered grid used in most 2-D overland flow models	44
Figure 3.4 (b)	Fully-dense grid used in this MacCormack time-splitting scheme	45
Figure 3.7.2 (a)	Fictitious points along boundaries	53
Figure 3.7.2 (b)	Reflection boundary condition used for slip boundaries when $\varepsilon = 0$	53
Figure 3.7.2 (c)	Non-slip boundary condition used when $\varepsilon \neq 0$	54
Figure 3.9	The procedure to assure a non-negative flow depth	59
Figure 3.13	A flowchart showing how the value of the sediment load, G , is updated	71

Figure 4.2 (a)	Sketch (not drawn to scale) of the 1D Steady State Problem with no Hydraulic Jump	73
Figure 4.2 (b)	The analytical solution to Test Problem 1	75
Figure 4.2 (c)	The free surface profile and bed level for test problem 1	75
Figure 4.3	The numerical steady solution is compared with the Analytical Steady Solution for test problem 1	77
Figure 4.4.2 (a)	Sketch (not drawn to scale) of the 1D Steady State Problem With an Hydraulic Jump	80
Figure 4.4.2 (b)	The Analytical Solution to Test Problem 2	82
Figure 4.4.2 (c)	The free surface profile and bed level for Test Problem 2	82
Figure 4.4.3	The Numerical Steady Solution compared with the Analytical Steady Solution	84
Figure 4.4.4	The Numerical Steady Solution with Artificial Viscosity compared with the Analytical Steady Solution	86
Figure 4.7	The influence of boundary conditions	91-92
Figure 4.8 (a)	Quiver plot of a flow whose spill source was turned off after a minute	94

Figure 4.8 (b)	Flow wave pattern after 15mins	95
Figure 4.8 (c)	The surface profile of the pollutant concentration at the end of the 15min simulation	96
Figure 4.9 (a)	Vertical infiltration profile in 3-D. The flow is over sand	98
Figure 4.9 (b)	Vertical infiltration profile in 3-D. The flow is over loam	98
Figure 4.9 (c)	Vertical infiltration profile in 3-D	99
Figure 4.11 (a)	A mesh plot of the elevation data obtained from the DEM of a gas drilling area in Central Arkansas	101
Figure 4.11 (b)	The free surface profile of the flow right after the start of the simulation ($t = 4s$)	103
Figure 4.11 (c)	The free surface profile of the flow right after 30s	104
Figure 4.11 (d)	A 2-D view of the mesh plot of the DEM showing the relative elevation of the cells	106
Figure 4.11 (e)	A 2-D view of the free-surface profile of the water depth after 30s of simulation	106
Figure 4.11 (f)	The flow field shows the magnitude and direction of the velocity	107
Figure 4.12 (a)	A 1-D plot of pollutant concentration profile	109

Figure 4.12 (b)	A mesh plot of pollutant concentration profile	109
Figure 4.13 (a)	Mesh plot of Sediment load in the x -direction, G_x	111
Figure 4.13 (b)	Mesh plot of Sediment load in the y -direction, G_y	111

LIST OF TABLES

Table 1.5	Fracturing Fluid Additives, Main Compounds, and Common Uses	6
Table 2.6.1	Some Earlier Equations for Flow and Resistance Coefficients	21
Table 2.6.2	Manning coefficient values, n , for some typical surfaces.	22
Table 2.7	Common Infiltration Models	26
Table 2.9.1	Examples of Erosion/sediment transport models	30
Table 2.9.2	Some commonly used algorithms for sediment transport in physics-based erosion models	32
Table 3.7	Minimum Required number of boundary conditions at open or 'ocean' boundaries	55
Table 3.11	Default Values for Soil Hydraulic Parameters	64
Table 4.9	Comparing the relative values of the pore-size distribution index with the numerical results given for the rate of infiltration and cumulative infiltration depth for three different soil textures.	82

NOMENCLATURE

Symbol	Meaning	Dimensions
α	Empirically determined parameter used to calculate mixing coefficients in the x -direction	1
β	Empirically determined parameter used to calculate mixing coefficients in the y -direction	1
γ	Specific Weight	$M/L^2/T^2$
ε	Eddy viscosity	L^2/T
ε_x	Eddy viscosity in the x -direction	L^2/T
ε_y	Eddy viscosity in the y -direction	L^2/T
ϵ	Multiplier used in artificial viscosity calculations	1
θ	Soil Moisture Content	$L^3 \text{ water}/L^3 \text{ soil}$
θ_i	Initial Soil Moisture Content	$L^3 \text{ water}/L^3 \text{ soil}$
θ_r	Residual Soil Moisture Content	$L^3 \text{ water}/L^3 \text{ soil}$
θ_s	Soil Moisture Content at Saturation	$L^3 \text{ water}/L^3 \text{ soil}$
κ	Constant used to regulate the amount of dissipation in artificial viscosity calculations	1
λ	Pore-Size Distribution Index	1
ν	Kinematic Viscosity	L^2T^{-1}
ρ	Density	M/L^3
τ_c	Critical Shear Stress	$M/L/T^2$
τ_f	Flow or Hydraulic Shear Stress	$M/L/T^2$
u_x	Normalized form of the Gradients of a variable in the x -direction	1

v_y	Normalized form of the Gradients of a variable in the y -direction	1
ψ_a	Air Entry Suction Head	L
ψ_f	Wetting Front Suction Head	L
Γ_x	One-dimensional x -operator	-
Γ_y	One-dimensional y -operator	-
Δt	Time step size	T
Δt_x	Fractional time step taken in the x -direction	T
Δt_y	Fractional time step taken in the y -direction	T
Δx	Length of the grid	L
Δy	Width of the grid	L
E_x	Momentum Source Terms in the x -direction	L^2/T^2
E_y	Momentum Source Terms in the y -direction	L^2/T^2
K_t	Sediment transport coefficient	$M^{-1/2}L^{1/2}T^2$
Λ	Artificial Viscosity Operator	1
T_c	Sediment transport capacity	$M/L/T$
c	Wave celerity	L/T
dl/dt	Infiltration Rate	L/T
h, H	Flow depth	L
h_{min}	Minimum value flow depth must be for a condition to kick in	L
C	Chezy coefficient	1
C	Pollutant Species Concentration	M/L^3
C_n	Courant number	1

C_s	Concentration of Sediment	M/L^3
D_f	Net detachment rate	$M/L^2/T$
D_i	Interrill sediment delivery to the rill	$M/L^2/T$
D_r	Rill net detachment or deposition rate	$M/L^2/T$
D_x	Mixing Coefficient in the x -direction	L^2/T
D_y	Mixing Coefficient in the y -direction	L^2/T
F	Darcy-Weisbach coefficient	1
F	Sum of convective acceleration and pressure force terms in the x -direction	L^3/T^2
g	Acceleration due to gravity	L/T^2
G	Sediment load	$M/L/T$
G	Momentum arising in one direction due to fluid movement in an orthogonal direction	L^3/T^2
G_x	Sediment load in the x -direction	$M/L/T$
G_y	Sediment load in the y -direction	$M/L/T$
I	Cumulative Infiltration Depth	L
I	Index to locate the x -position of a cell	-
J	Index to locate the y -position of a cell	-
K_s	Soil Hydraulic Conductivity	L/T
N	Manning coefficient	1
R	Spill Rate	L/T
S	Sum of convective acceleration and pressure force terms in the y -direction	L^3/T^2
S_{fx}	Friction slope in the x -direction	1
S_{fy}	Friction slope in the y -direction	1

S_{ox}	Bed slope in the x -direction	1
S_{oy}	Bed slope in the y -direction	1
U	Unit width discharge in the x -direction	L^2/T
u	velocity in the x -direction	L/T
V	Unit width discharge in the y -direction	L^2/T
v	velocity in the y -direction	L/T
Z	Bed elevation	L
z	Vertical axis	L
$\frac{\partial y}{\partial x}$	Partial derivative of y with respect to x	-
c	Superscript denoting the variable is the corrected value calculated in the Corrector sequence of the MacCormack scheme	-
o	Superscript denoting the variable is the initial value or old value calculated from the previous time step in the MacCormack scheme	-
p	Superscript denoting the variable is the predicted value calculated in the Predictor sequence of the MacCormack scheme	-

ABBREVIATIONS

Abbreviation	Full Meaning
ANL	Argonne National Laboratory
BMP	Best Management Practice
BRIC	Brazil, Russia, India and China
CAST	Center for Advanced Spatial Technology
CFD	Computational Fluid Dynamics
DEM	Digital Elevation Model
EOR	Enhanced Oil Recovery
EPA	Environmental Protection Agency
Eqn.	Equation
FD	Finite Difference
FEM	Finite Element Method
FSP	Fayetteville Shale Play
GIS	Geographic Information System
IPAS	Infrastructure Placement Analysis System
LHS	Left Hand Side
LINGO	Low Impact Natural Gas and Oil
MOC	Method of Characteristics
NORM	Naturally Occurring Radioactive Material
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
RHS	Right Hand Side

SWAT	Soil Water Assessment Tool
SWE	Shallow Water Equation
USGS	United States Geological Survey
WEPP	Water Erosion Prediction Project
1-D	One Dimensional
2-D	Two Dimensional
3-D	Three Dimensional

CHAPTER 1

INTRODUCTION

1.1 World Energy Outlook

As the world crosses the 7 billion population mark, energy is needed more than ever. Energy is needed to grow food, provide drinkable water, power industries and sustain societies. The rise of the middle class in BRIC (Brazil, Russia, India and China) countries also places an unprecedented demand on the world's supply of energy (Bhar and Nikolova, 2009; Tamazian et al, 2009). One way to meet this demand is to develop renewable sources of energy. Energy sources such as wind, solar, biofuels and geothermal are called renewable because they can be replenished as opposed to fossil fuels that cannot be replaced once withdrawn. But renewable sources of energy have not gained as much traction as fossil fuels because they often do not meet the basic criteria of energy supply which is that in addition to being clean, it must be cheap, abundant and reliable (Friedman, 2008). Renewable sources of energy leave a lower carbon footprint than fossil fuels but until the economics of the process makes sense it is unlikely to replace the world's dependence on fossil fuels. A study by Royal Dutch Shell estimates that renewable sources of energy would make up at most 30% of total world primary energy by 2050 while fossil fuels would be at least 55% (Shell, 2011).

Two other ways the high demand for energy is being met are intensified exploration efforts and improvements in the production rates of existing wells. The energy industry is now actively drilling in frontier areas – for instance, ultra-deep water and the Arctic – which

they have shunned for a long time. Enhanced Oil Recovery (EOR) techniques enable existing wells to recover more of their reserves and produce for longer periods.

1.2 Natural Gas

Natural gas is now virtually tied with coal – both contribute 22% – as the United States' largest domestic energy resource. It accounts for 25% of the US energy consumption, second only to petroleum (AEO, 2011). Natural gas is mostly methane (CH₄) and it is the cleanest burning fossil fuel. It has risen in importance because in addition to being clean, it is also abundant, cheap and reliable. Once liquefied, its volume shrinks to about 1/600 of its original size and so vast amounts of it can be transported just as easily as, if not as safely, oil (Vargaftik, 1975; Thomas and Dawe, 2002).

1.3 Conventional vs. Unconventional Gas

Natural gas that occurs in discrete oil and gas reservoirs is termed “conventional” gas. It is, just like oil, trapped in porous sedimentary rocks (Compton, R.R., 1985). It is produced by drilling vertical wells into the structural or stratigraphic traps which hold it in place. The majority of gas produced in the US is conventional gas (AEO, 2011).

Gas found in other geologic formations is termed “unconventional.” There is no universal definition for unconventional formations (Coleman, 2011). They are generally defined as gases that are produced or extracted with techniques other than the traditional vertical drilling of wells. Examples of unconventional gases include: are shale gas, tight gas sands,

coalbed methane and Arctic and sub-sea hydrates (Johnson and Dore, 2010). Shale gas forms the bulk of unconventional gas produced in the US because it is found in vast amounts and also because favorable of economics and advanced drilling technology. The technology is horizontal drilling and hydraulic fracturing.

1.4 Hydraulic Fracturing

Hydraulic Fracturing is the process where large amounts of water under high pressure are forced down a well to crack or fracture gas-bearing shales so that the rock can release its gas (Howarth et al, 2011). The amount of water used depends on the nature of the sub-surface. A well can require between 2 and 8 million gallons of water for a fracturing job. 10,000 pounds per square inch is the typical level of pressure needed to fracture a shale rock (Linkov, 2012).

Fracturing is not new. The first fracture job dates back to 1858 – a mere 23 years after the first recorded US gas well was drilled. The fracturing was done then using black powder as explosives. It would take close to a century however before engineers in Halliburton working out of Velma, OK in 1949 could use water to fracture rocks. There was a marked increase in the use of hydraulic fracturing in the 1980s when it was successfully combined with horizontal drilling in the Barnett Shale (Gidley et al, 1990 and Cooke et al, 2010).

Hydraulic fracturing is essentially an EOR technique. It can be used to stimulate production in wells experiencing declining output. Its promise though is that it can also be used to produce gas from formations that would not yield to conventional methods. Hydraulic

fracturing is a costly process. Cheap gas prices in the 1990s made it uneconomical to exploit for gas at that time. Higher gas prices in the mid-2000s and improved horizontal drilling techniques are factors that made hydraulic fracturing a practical approach to exploiting natural gas. As a process, it has delivered its promises. It is currently the topic that receives the most attention from the academy, industry and government in energy outlook discussions (see for example, Williamson, 2012; Egan, 2012 and Osborn, 2012).

One major concern of the public about hydraulic fracturing is the surface impacts on the environment (Cupas, 2009). Hydraulic fracturing often takes place within city limits and residents want to know how these drilling activities are going to impact their land, water and way of life.

1.5 Reserve Pit Failures

1.5.1 What is a Reserve Pit?

Figure 1.1 shows the picture of a site that has been prepared for hydraulic fracturing. A reserve pit can be seen close to the drilling rig. A reserve pit, sometimes called a mud pit or an earthen pit, is an excavation near a well. It can be used to store drilling fluids prior to a fracture job but it is more commonly used to store drilling waste following a fracture job. In the case of a blowout of a wellhead, it can also serve as an initial containment of the spilled drilling mud (Leuterman et al, 1988). The size of reserve pits varies. Their sizes depend on the amount of fluid needed to drill a well. A deeper well requires more drilling fluid than a relatively

shallow one. The average volume for a reserve pit can range from 3,600 barrels for a well that is less than 4,000ft deep to 15,000 barrels for a well that is deeper than 15,000ft (USOTA, 1992).

1.5.2 Contents of a Reserve Pit

A reserve pit stores drilling fluids and produced water. The drilling fluids used consist of water, sand and chemical additives. Water and sand make up nearly 98% of a typical drilling fluid. Water is the most common drilling base or carrier fluid but oil or a synthetic compound can also be used where water fails to perform well. Water can even cause the bore to collapse in some shales. The chemical additives optimize fluid flow and help prevent corrosion, formation of scales and scouring (Fink, 2003). Examples of the main compounds used as additives, their purpose as a fracturing fluid additive and their everyday use is given in Table 1.5. Some of the contents of the drilling fluids are not published because oil servicing companies manufacturing and using them see it as a trade secret. There is increasing pressure on the industry however to disclose all chemicals used during hydraulic fracturing.

Produced water is water coming out of a well during production. This is water that has been in close contact with the formation for centuries and it contains a lot of brine and possibly hydrocarbons. Some contain naturally occurring radioactive material (NORM) (Veil et al, 2004). Flowback water is also sometimes loosely referred to as produced water. Flowback water is the water that “flows back” after being injected under high pressure to crack a well. The amount of flowback water

Table 1.5: FRACTURING FLUID ADDITIVES, MAIN COMPOUNDS, AND COMMON USES

Adapted from Shale Gas Primer, GWPC (2009)

Additive Type	Main Compound(s)	Purpose
Diluted Acid (15%)	Hydrochloric acid or muriatic acid	Help dissolve minerals and initiate cracks in the rock
Biocide	Glutaraldehyde	Eliminates bacteria in the water that produce corrosive byproducts
Corrosion Inhibitor	N,n-dimethyl formamide	Prevents the corrosion of the pipe
Crosslinker	Borate salts	Maintains fluid viscosity as temperature increases
Gel	Guar gum or hydroxyethyl cellulose	Thickens the water in order to suspend the sand
Surfactant	Isopropanol	Used to increase the viscosity of the fracture fluid

depends on the geology of the area; usually between 30% and 70% of the fracturing fluid can return as flowback water (Gupta and Hlidek, 2009). The flowback water thus contains all the constituents of a drilling fluid plus other chemicals it comes in contact with in the well. These chemicals vary from well to well and an exhaustive list of what may be in flowback water is currently not available.

1.5.3 Failure of Reserve Pits

Hydraulic fracturing is a water intensive procedure and so its associated reserve pit usually contains hundreds of thousands of gallons of drilling mud and flowback water. These earthen-walled pits can fail. Improper pit designs, poor maintenance and wrong pit locations are factors that can cause failure (EPA, 2000). During a storm event, for instance, snowmelt or rain fall exceeding 0.5 inches, reserve pits can also overflow. An outright failure can cause up to 100,000 gallons of toxic fracturing fluids to be spilled. When these happen, the physical, chemical and biological integrity of both land and nearby waters are compromised. Some of the chemicals used in drilling have known harmful health effects, for instance, benzene – a chemical additive – causes cancer (Cox and Ricci, 2006). Even if the chemicals used are not harmful, their non-disclosure has the public concerned. There is the fear of the unknown and the public do not trust that the companies will protect them.



Figure 1.1: A drilling site where hydraulic fracturing is taking place. A reserve pit the size of a swimming pool is seen adjacent to the drilling rig. The trucks pump in millions of gallons of fracturing fluid under high rate to crack the dense organic gas-bearing shale rock several thousands of feet below. Drilling mud is stored in the reserve pit. If the reserve pit overflows due to excessive rainfall or one of its earthen walls collapses, there can be severe environmental consequences. *(Photo courtesy: Tom Hayes of Gas Technology Institute)*

1.6 Fayetteville Shale Play

The Fayetteville Shale Play (FSP) is an unconventional natural gas resource that lies in Central Arkansas (Figure 1.2). It was discovered in 2002 by geologists from Southwestern Energy and has since become the nation's third largest producing shale play (Southwestern, 2003). It is one of the six active shales in the US. The other producing shales in the US to date are the Barnett Shale, Marcellus Shale, New Albany Shale, Antrim Shale and Haynesville Shale. Large scale development of the FSP began in 2005 and production has grown rapidly since then (Figure 1.2). Less than 50 wells were drilled in 2005; by the

end of 2010, over 3,000 wells had been drilled (Powers, 2011). This rapid increase in the number of wells drilled and the associated number of reserve pits that are being constructed has the public worried about what the consequences of a large spill would be and what safety measures are in place to deal with such incidents. News of polluted springs, wetlands and water wells resulting in death of animals and damaged vegetation abound in the media. This project aims to address the concerns of the major stakeholders in the FSP and provide a sustainable solution to the problem of a reserve pit failure and its consequences.

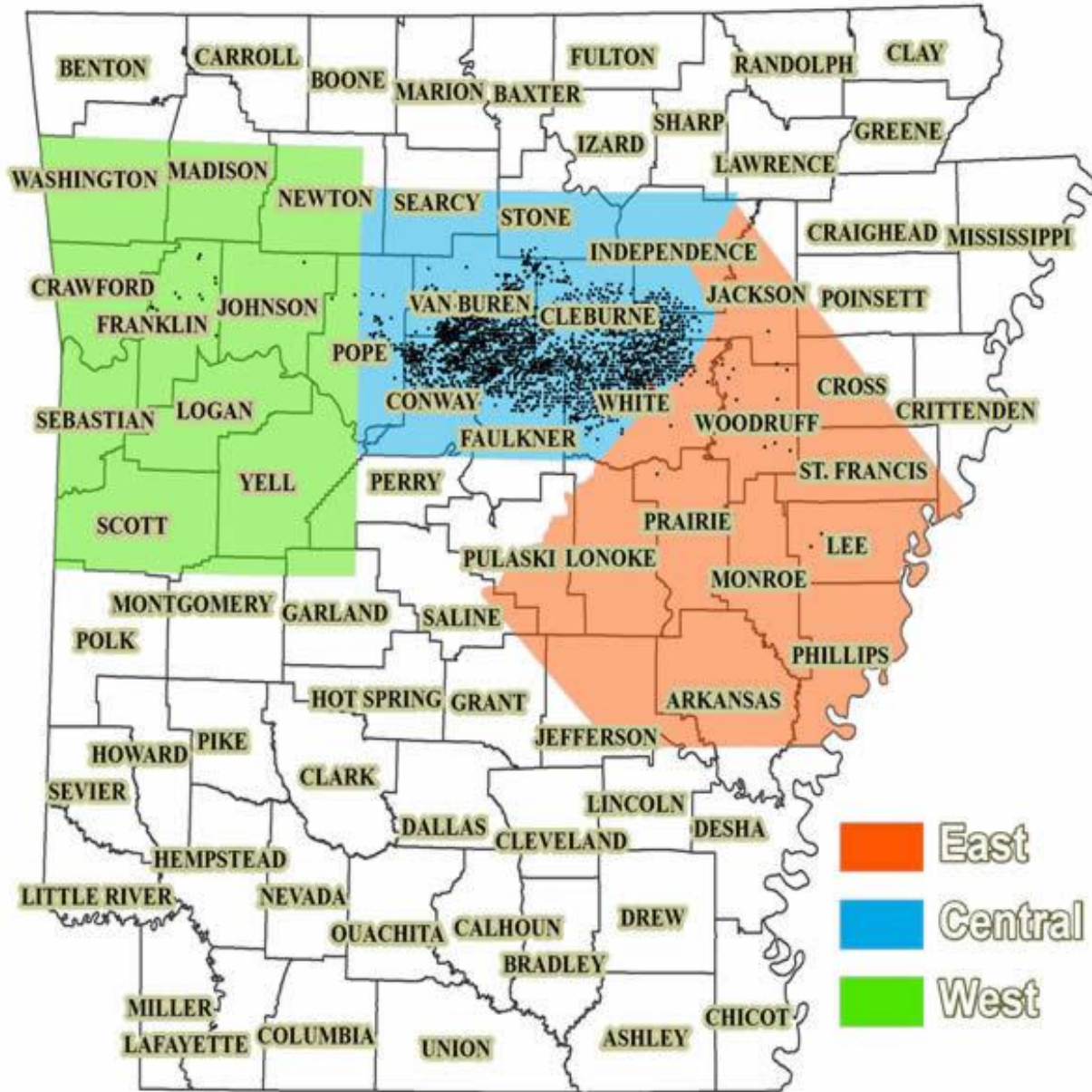


Figure 1.2: Producing wells in Arkansas by location. A black dot represents a producing well. Most of the development of the FSP has taken place in Central Arkansas.

Source: Arkansas Geology Survey,

http://www.geology.ar.gov/fossilfuel_maps/fayetteville_play.htm Accessed on 02/06/12

1.7 Aim

The aim of this work is to develop a sustainable solution to the potential environmental impact of a spill. A sustainable solution would make economic sense, be environmentally friendly and socially responsible. It has to be cost-effective to the companies that would be implementing it, be a safer alternative to current approaches and be transparent to the public so that they are in the know. This work does all that and more.

A mathematical model is developed to simulate the overland flow of a spilled fracturing fluid. The model uses a minimum of input and is computationally fast enough to make its use as a real time screening tool justifiable. The model will be integrated into the Infrastructure Placement Analysis System (IPAS) where it will serve as a decision support system for key stakeholders. The IPAS is a Geographic Information System (GIS) based platform created by the Low Impact Natural Gas and Oil (LINGO) project. LINGO is an initiative of a partnership between the University of Arkansas' Chemical Engineering Department, Center for Advanced Spatial Technology (CAST) and Argonne National Laboratory (ANL). The IPAS is a web-based resource. The website also contains information about the Fayetteville Shale and resources the public can use to acquaint themselves with the process. Fear of the unknown risks caused by the exploration and exploitation of gas is a major reason for the concerns raised by the public; it is the aim of this work to address those fears. The FSP is the case study for this work. The result of this project can be applied – with some modifications – in other shale basins.

1.8 Objectives of this Project

This project will answer the question of where do I put my reserve pit? Should it be to the left or right of the well pad? How far should it be from a water body or a protected species? The project will help decision makers in screening for suitable locations to site a reserve pit. A suitable location is one that has the least potential impact on the environment in the case of a reserve pit failure.

This work is going to be a piece of the LINGO IPAS. The overall decision support system will help not only in siting reserve pits but in screening for the best sites to prepare for drilling. By reducing the number of detailed physical on-site surveys that need to be carried out before deciding on an area for gas drilling, this work should save gas producers both time and money.

The results from this work will also help evaluate the cost associated with the Best Management Practices (BMPs) of reserve pit construction. Questions such as:

1. Should a double berm be constructed?
2. Do you take the risk of siting the well and its reserve pit in an area?
3. What additional control should be put in place?

are answered from a practical standpoint. For instance, if the clean-up cost in an area will be minimal and no protected class of species is going to be affected in the case of a spill, a gas producer may decide to go ahead and drill and put in a reserve pit. This work can help such a producer know this ahead of time.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The 2010 BP *Deepwater Horizon* oil spill at the Gulf of Mexico has renewed the public interest in the safety of oil and gas exploration and production process. The Macondo oil well blowout disaster has already eclipsed the (in-) famous *Exxon Valdez* oil spill of 1989 both in terms of volume spilled and the environmental impact (Sylves & Comfort, 2012). There is no dearth of oil spills models in the literature. Oil spill models are usually developed for use in marine environments (Fingas, 2010); to the best of the author's knowledge, there is no oil spill model for use on land.

Onshore spill models are rare to come by whether the fluid of concern is oil, fracturing fluids or drilling mud from a reserve pit failure. In order to develop a spill model for the LINGO IPAS, physical relationships would be developed from the fundamental principles of mass and energy balance.

2.2 The Shallow Water Equations (SWEs)

At the core of our flow model is a set of governing equations referred to as the Shallow Water Equations (SWEs). They are sometimes called the St. Venant equations after the late 19th century French mathematician who first derived them (see St. Venant, 1871). Motion of objects is guided by Newton's 2nd law. If we apply Newton's 2nd law to the motion of fluid particles and make assumptions such as fluid stress is proportional to the gradient of

velocity and add a pressure term, we arrive at the Navier-Stokes equations. The Navier-Stokes equations completely describe the motion of fluids in three dimensions (3-D). Solution to the Navier-Stokes equation gives the velocity field. Once the velocity field is found, other quantities of interest such as the drag force or the mass flux can be determined. The Navier-Stokes equation is a numerically challenging problem to solve and that is why Computational Flow Dynamics (CFD) software packages are written for it. We can customize the Navier-Stokes equation for overland flow by making some simplifying assumptions such as (Lai, 1986, Strelkoff, 1970; Iwasa, 1988; Zhang and Cundy, 1989):

1. Hydrostatic pressure distribution i.e. the vertical acceleration of a fluid particle is very small compared with the acceleration of gravity g and hence can be neglected;
2. The density of the water is fairly constant throughout (incompressible);
3. The bed slope is small, small enough to make the assumption of $\sin \theta \approx \cos \theta$ a valid one;
4. The channel bottom is immovable or relatively stable and so fixed with respect to time;
5. Only shear stresses arising from the horizontal components of the velocity are significant and therefore considered, shear stresses due to the vertical velocity components are neglected and
6. The coefficient of hydraulic resistance for unsteady flow is the same as that for steady flow, and so it can be estimated from Darcy-Weisbach, Chezy or Manning equations.

The St. Venant equations result when the 3-D Navier Stokes equations are vertically averaged over depth in their full dynamic form, the St. Venant equations can be written as (Garcia and Kahawita, 1986):

$$\frac{\partial h}{\partial t} + \frac{\partial(uh)}{\partial x} + \frac{\partial(vh)}{\partial y} = R - \frac{dI}{dt} \quad (2.1)$$

$$\frac{\partial(uh)}{\partial t} + \frac{\partial\left(u^2h + \frac{1}{2}gh^2\right)}{\partial x} + \frac{\partial(uvh)}{\partial y} = gh(S_{ox} - S_{fx}) + \text{other terms} \quad (2.2)$$

$$\frac{\partial(vh)}{\partial t} + \frac{\partial(uvh)}{\partial x} + \frac{\partial\left(v^2h + \frac{1}{2}gh^2\right)}{\partial y} = gh(S_{oy} - S_{fy}) + \text{other terms} \quad (2.3)$$

where $h = h(x, y, t)$; $u = u(x, y, t)$ and $v = v(x, y, t)$ are the fluid depth, velocity in the x -direction and velocity in the y -direction respectively. Equation 1 is a statement of mass conservation while Eqns. 2 & 3 are derived from the momentum conservation principle. Other terms such as diffusion, Coriolis acceleration and wind-induced shear stresses can be added to the right hand side (RHS) of the momentum equations. See Appendix A for the derivation of the St. Venant equation from first principles.

The equations allow us to compute the flow depth and flow velocity field of an area under unsteady precipitation and infiltration and given realistic upstream and downstream boundary conditions. Knowing the flow depth and velocity field can help answer questions such as: How much discharge can we expect from overland sections of an area? How much sediment and/or pollutant will be washed off and where will they be deposited? What does the time-space evolution of transported pollutants look like in surface waters? More

importantly, the model can help us run what-if scenarios. Being a physical model, it will enable us to see how the relationship between topography, bed roughness, soil hydraulic properties, man-made features and rainfall impact the hydrologic and hydraulic behavior of an area.

2.3 Solving the SWEs

The first serious attempts to apply the SWEs to practical hydrologic and hydraulic problems were made when computers came to be used for numerical computation. Before then, analytical and graphical techniques were employed. Analytic solutions have been restricted to limited regions of the solution domain or to special cases where suitable simplifications can be made. Even for these special cases, graphical techniques are prohibitively slow (e.g. Massau, 1889). The full SWEs are too complicated to be solved analytically hence approximate forms have been developed. These simplifications are used when physical field parameters show that accuracy is not seriously compromised (Govindaraju, 1988a). Two main models of the St. Venant equations arose from such efforts: the kinematic wave model and the diffusive wave model.

2.4 Kinematic Wave Model

Henderson and Wooding (1964) developed the kinematic wave model by equating the friction slope to the bed slope and neglecting all other terms in the momentum equations.

In the Kinematic Wave approximation: $S_{ox} = S_{fx}$ and $S_{oy} = S_{fy}$ therefore Eqns. 1 - 3 becomes:

$$\frac{\partial h}{\partial t} + \frac{\partial(uh)}{\partial x} + \frac{\partial(vh)}{\partial y} = R - \frac{dI}{dt} \quad (2.4)$$

$$\frac{\partial(uh)}{\partial t} + \frac{\partial\left(u^2h + \frac{1}{2}gh^2\right)}{\partial x} + \frac{\partial(uvh)}{\partial y} = 0 \quad (2.5)$$

$$\frac{\partial(vh)}{\partial t} + \frac{\partial(uvh)}{\partial x} + \frac{\partial\left(v^2h + \frac{1}{2}gh^2\right)}{\partial y} = 0 \quad (2.6)$$

This huge simplification of the SWEs made it amenable to analytical solution. Using the Method of Characteristics (MOC), Woolhiser and Liggett (1967) showed that for a rising hydrograph, the kinematic wave model approaches the solution for the full St. Venant equation. Other workers (e.g. Woolhiser, 1974; Hjelmelt, 1981; Parlange et al, 1981) extended the scope of overland flow problems that can be solved using the kinematic wave assumption. All these solutions are based on the MOC so it is not possible to obtain analytical solutions when the characteristic curves intersect within the flow domain. Therefore, the kinematic wave model fails when spatially non-uniform rainfall or infiltration rate or initial conditions cause variations in the wave speed and make these characteristic curves meet. Mathematics aside, the kinematic wave model is simply not valid for highly subcritical flows and for flows where the downstream boundary is an important factor.

2.5 Diffusion Wave Model

To overcome the shortcomings of the kinematic wave model, Morris and Woolhiser (1980) introduced the diffusion wave model. The diffusion wave approximation of the SWEs uses the full continuity equation but simplifies the momentum equations by considering the friction slope to be the sum of bed slope and flow depth gradient and neglecting other terms at the RHS of Eqns. 2 & 3.

In the diffusion wave approximation: $S_{fx} = S_{ox} - \frac{\partial h}{\partial x}$ and $S_{fy} = S_{oy} - \frac{\partial h}{\partial y}$ so that the RHS of Eqns. 2 & 3 can be written as

$$gh(S_{ox} - S_{fx}) = gh \frac{\partial h}{\partial x} \text{ or } \frac{\partial(1/2gh^2)}{\partial x} \text{ and } gh(S_{oy} - S_{fy}) = gh \frac{\partial h}{\partial y} \text{ or } \frac{\partial(1/2gh^2)}{\partial y}.$$

Equations 1-3 thus become:

$$\frac{\partial h}{\partial t} + \frac{\partial(uh)}{\partial x} + \frac{\partial(vh)}{\partial y} = R - \frac{dI}{dt} \quad (2.7)$$

$$\frac{\partial(uh)}{\partial t} + \frac{\partial(u^2h)}{\partial x} + \frac{\partial(uvh)}{\partial y} = 0 \quad (2.8)$$

$$\frac{\partial(vh)}{\partial t} + \frac{\partial(uvh)}{\partial x} + \frac{\partial(v^2h)}{\partial y} = 0 \quad (2.9)$$

The diffusion wave model performed as well as the kinematic wave model in all the cases studied by Ponce et al (1978). The model is superior to the kinematic wave model in that it allows for physical attenuation and hence better able to handle the dissipative tendencies

found in the full dynamic form. The diffusion wave model also has the added advantage of being able to compute backwater effects i.e. it allows backfacing slopes in flow fields.

However, by neglecting convective acceleration terms in the momentum equation, both the kinematic and diffusion wave model have limited practical application. We will be solving the full dynamic form of the St. Venant equations so that no important terms will be left out.

2.6 Hydraulic Resistance Parameters

One important aspect of simulating the SWEs is the description of velocity of flow which is often described by resistance equations (Horton, 1938). Some earlier equations for flow and resistance coefficients are given in Table 2.6.1. The resistance to flow in these equations is typically caused by the soil surface, vegetative cover and man-made structures and it is accounted for by a coefficient of hydraulic resistance. A reliable estimation of this coefficient is difficult since its value varies with the type and density of soil cover, the condition of the surface, and the flow depth relative to bed roughness (Christensen, 1985). Thus, the relation derived for a site may not be generally applicable for other sites and may not even be applicable at the same site when some flow conditions changes.

A large number of studies (e.g. Williams, 1970; Turner, 1978; Sweeten, 1969) have been carried out so that a general understanding of the variation of hydraulic resistance has been developed. The studies can generally be classified into three: empirical, theoretical and semi-empirical (ASCE, 1963 and Kadlec, 1990). Experience shows that the best

resistance equations are semi-empirical: they are grounded in hydrodynamic principles and have parameters that need to be determined experimentally (Schmitz, 1985).

The Chezy equation, Manning equation and the Darcy-Weisbach equation are the common resistance equations in use. In all the equations, an empirical coefficient is defined which is not constant but depends on the shape of the channel cross-section, boundary roughness, flow velocity and depth (Manning, 1889). Their coefficients are related as:

$$\frac{C}{g^{1/2}} = \frac{R^{1/2}}{ng^{1/2}} = (8/f)^{1/2} \quad (2.10)$$

where C , n and f are the Chezy, Manning and Darcy-Weisbach coefficients respectively, R is the hydraulic radius and g is the acceleration due to gravity. The Chezy equation is more applicable to flows in rivers and other deep water flows where the assumption of the resisting force being proportional to the square of velocity holds true (Kao, 1978). We define deep water flows as flows where the ratio of fluid depth to equivalent roughness height is greater than 20 (Ree et al, 1977). Overland flow typically involves shallow water movements and so I did not use the Chezy equations. The Darcy-Weisbach equations are most suitable for laminar flows and are not the best choice for transitional (laminar/turbulent) flows commonly encountered in practice. The Manning equation has found widespread acceptance in hydrology because it can deal with laminar, transient and turbulent flows (Dooge, 1989). Consequently, the Manning coefficient, n , has been estimated for a number of surfaces (Table 2.6.2).

Table 2.6.1: Some Earlier Equations for Flow and Resistance Coefficients (Source: Maheshwari, 1992)

Formula Name	Year	Equation	Source
Chezy	1768	$V = C (RS_e)^{1/2}$	Chow (1959)
Darcy-Weisbach	1845	$(8/f)^{1/2} = V/(RgS_e)^{1/2}$	Henderson (1966)
Ganguillet and Kutter ¹	1869	$C = \frac{23 + \left(\frac{0.00155}{S_e}\right) + (n_k)^{-1}}{1 + [23 + \left(\frac{0.00155}{S_e}\right)]^{n_k} / R^{1/2}}$	Chow (1959)
Manning	1891	$V = R^{2/3}S_e^{1/2}/n$	Manning (1889)
Pavlovsky ²	1925	$C = n^{-1}R^y$	Chow (1959)
Colebrook and White ³	1938	$\left[\frac{1}{f^{1/2}}\right] = -C \ln\left[\frac{k}{a_1 R} + \frac{a_2}{R_N f^{1/2}}\right]$	ASCE (1963)

¹Where n_k is Kuttler's resistance coefficient

²Where $y = 2.5n^{1/2} - 0.13 - 0.75R^{1/2}[n^{1/2} - 0.10]$

³Where a_1 and a_2 are constants

Table 2.6.2: Manning coefficient values, n , for some typical surfaces. Values were compiled from Cowan (1956), Chow (1959), Henderson (1966) and Chaudhry (1993)

Material/Surface	n	Material/Surface	n
<i>Natural Streams</i>		<i>Metals</i>	
Rivers	0.030	Brass	0.011
Deep Pools	0.040	Cast Iron	0.013
Straight and Clean	0.035	Smooth Steel	0.012
		Corrugated Metal	0.022
<i>Floodplains</i>		<i>Non-Metals</i>	
Pasture, Farmland	0.035	Glass	0.010
Light Brush	0.050	Clay Tile	0.014
Heavy Brush	0.075	Brickwork	0.015
Trees	0.15	Asphalt	0.016
		Masonry	0.025
		Finished Concrete	0.012
<i>Excavated Earth Channels</i>		Unfinished Concrete	
Clean	0.022	Gravel	0.029
Gravelly	0.025	Earth	0.025
Weedy	0.030	Planed Wood	0.012
Stony, Cobbles	0.035	Unplaned Wood	0.013

It must be kept in mind that these estimates are just that – an educated guess. As has been emphasized, any hydraulic coefficient relies on a number of factors and surface or bed roughness is just one factor. Arcement and Schneider (2012) provide a guide on what value of n to choose as well as a comprehensive review of the Manning equation.

Since we are not using either the kinematic wave or diffusion wave approximation, the friction slope or energy, S_f , has to be calculated. This is where having a value for n is needed to calculate the energy slope using the following relation:

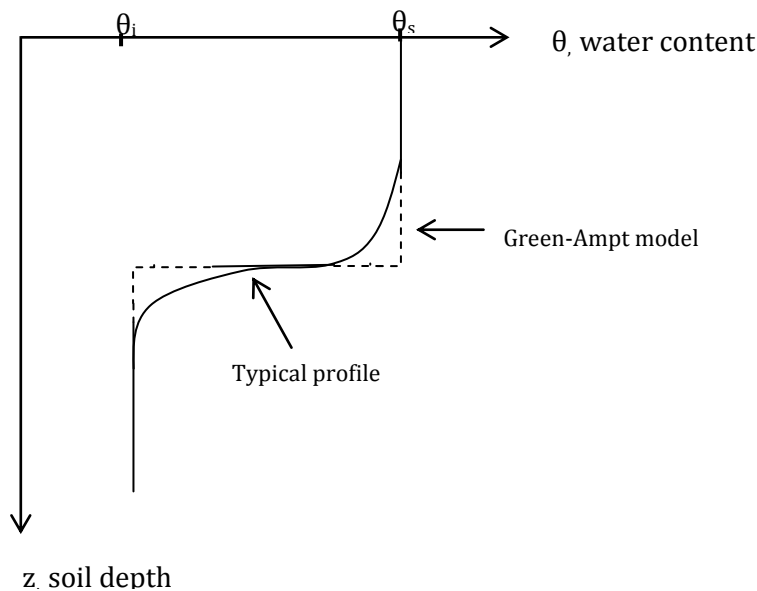
$$S_{fx} = \frac{n^2 u (u^2 + v^2)^{1/2}}{h^{4/3}} \quad (2.11)$$

and

$$S_{fy} = \frac{n^2 v (u^2 + v^2)^{1/2}}{h^{4/3}} \quad (2.12)$$

2.7 Infiltration Model

The RHS of the continuity equation (Eqn. 2.1) in the SWEs contain the rate of infiltration term, dI/dt . Infiltration is the term used to describe the process of water entry into the soil and the rate of infiltration is the flux passing through soil surface and flowing into the profile (Hillel, 1971). The infiltration rate is initially high if the soil is dry but as water supply to the surface continues, the rate reduces until it tends to the hydraulic conductivity of the soil (Phillip, 1973). A typical infiltration moisture profile is shown in Figure 2.7. Heber Green and Gussy Ampt were among the first to relate soil properties to the rate at which water infiltrates. The now famous Green-Ampt (1911) model assumes an instantaneous change in water content at the wetting front from a uniform θ_i to the water content at natural saturation, θ_s (Figure 2.1)



θ_i - Initial Moisture

θ_s - Saturated Moisture Content

Figure 2.7: A Water-Content vs. Depth curve. A typical infiltration profile is shown. The Green-Ampt model is superimposed to show how their assumption of instantaneous change in moisture content deviates from reality. The model is useful though for most practical purposes.

The physics of soil water movement is still an area of active research (Ebel et al, 2012). In the century following the publication of the Green-Ampt model, many researchers have sought to deepen our understanding of the phenomenon by either showing the limits of the Green-Ampt model (e.g. Ghosh, 1980; Youngs, 1987; Swartzendruber & Youngs, 1974), developing better ways to estimate its parameters (e.g. Rawls et al, 1981; Morel-Seytoux & Khanji, 1974; Brakensiek, 1977) or coming up with a new infiltration model altogether (e.g. Richards, 1931; Horton, 1940; Barry et al, 1995). Table 2.7 shows some of the common infiltration models in use today.

Clausnitzer et al (1998) compared the precision and accuracy of estimated parameter confidence intervals of common infiltration models. On an order-of-magnitude basis, the level of accuracy of these models is virtually the same whereas the more sophisticated ones require higher computing cost. The simple Green-Ampt model as modified by Smith and Parlange (1978) is thus adequate for our purpose. Groves (1989) expressed this modified model as:

$$\frac{dI}{dt} = \frac{K_s}{1 - \exp\left(-\frac{I}{\Delta\theta\psi_f}\right)} \quad (2.13)$$

where $\Delta\theta = (\theta_s - \theta)$ and ψ_f is the suction head at the wetting front, θ is soil water content, θ_s is soil water content at saturation, I is the cumulative infiltration depth and K_s is the hydraulic conductivity of the soil.

Table 2.7 Common Infiltration Models (Modified from Clausnitzer et al., 1998)

Model	Equation	Classification	Parameters to fit
Horton (1940)	$I = \alpha_1 t + \frac{\alpha_2}{\alpha_3} [1 - \exp(\alpha_3 t)]$	Empirical	$\alpha_1, \alpha_2, \alpha_3$
Mezenecev (1948)	$I = \beta_1 t + \frac{\beta_2}{1 - \beta_3} t [1 - \beta_3]$	Empirical	$\beta_1, \beta_2, \beta_3$
Philip (1957c)	$I = At + St^{0.5}$	Semi-Analytic	A, S
Swartzendruber (1987)	$I = K_1 t + \frac{S}{A_0} [1 - \exp(-A_0 t^{0.5})]$	Semi-Analytic	K_1, S, A_0
Green and Ampt (1911)	$I = K_1 t + (h_s - h_f) \Delta \theta \ln \left(1 + \frac{I}{(h_s - h_f) \Delta \theta} \right)$	Physical/ Non-empirical	K_1, G
Parlange et al (1982)	$t = \frac{S^2}{2(\Delta K)^2(1 - \partial)} \left[2 \frac{\Delta K}{S^2} I - \ln \frac{\exp \left(2\partial \frac{\Delta K}{S^2} I \right) + \partial - 1}{\partial} \right]$	Physical/ Non-empirical	$\Delta K, S, \partial$
Barry et al (1995)	$I = K_i t + \frac{S^2 + 2K_1 h_s \Delta \theta}{2\Delta K(1 - \partial)} \left(t^* + 1 - \gamma - \exp \left(\frac{-6(2t^*)^{0.5}}{-6 + (2t^*)^{0.5}} - \frac{2t^*}{3} \right) + \frac{\gamma}{1 + t^*} \left\{ \exp \left(\frac{2t^*}{3} \right) [1 - (1 - \gamma)^8 t^{*2.5}] + (2\gamma + t^*) \ln \left(1 + \frac{t^*}{\gamma} \right) \right\} \right)$	Physical/ Non-empirical	K_i, K_1, B_1, B_2

$$G = (h_s - h_f) \Delta \theta$$

$$B_1 = (h_s + h_a) \Delta \theta$$

$$B_2 = \frac{2}{S^2 + 2K_1 h_s \Delta \theta}$$

$$t^* = \frac{2t(\Delta K)^2}{S^2 + 2K_1 h_s \Delta \theta}$$

$$\gamma = \frac{2K_1(h_s + h_a)\Delta\theta}{S^2 + 2K_1 h_s \Delta \theta}$$

2.8 Pollutant Transport Model

The hydrodynamic transport of pollutants is the result of the interaction between differential advection and turbulent diffusion both of which are dependent upon the flow velocity field (Cunge et al, 1980). The solution to the SWEs gives the flow velocity field. A pollutant transport model is therefore coupled to the SWEs to show the temporal and spatial evolution of concentration in the flow. Advection is the more dominant transport mechanism for most natural flows (Liang et al, 2010). The model is an advection-diffusion equation derived (see Appendix B for its derivation) from the principle of mass conservation and Fick's law.

Guymer et al (2005) renders this relationship as:

$$\frac{\partial C}{\partial t} + \frac{\partial(uC)}{\partial x} + \frac{\partial(vC)}{\partial y} = \frac{\partial}{\partial x} \left(D_x \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left(D_y \frac{\partial C}{\partial y} \right) + \text{source/sink term} \quad (2.14)$$

where C is the concentration of the pollutant and D_x and D_y are the mixing coefficients.

Pollutant transport models vary in the way D_x and D_y are determined and in what the sources and/or sinks are. The mixing coefficients are a combination of the depth-averaged longitudinal dispersion coefficients, turbulent diffusion coefficients and molecular diffusivities. Some models (e.g. Rodi, 1984) neglect molecular diffusion altogether because it is usually several orders of magnitude smaller than turbulent diffusion while others (e.g. Falconer, 1991) include it because it may be significant if flow turns laminar. Both the depth-averaged longitudinal dispersion coefficients and turbulent diffusion coefficients are dependent on flow properties (Hayter & Pakala, 1989) and there are physical models available to determine them separately (Huang, 2009). Separate estimation of the depth-

averaged longitudinal dispersion coefficients, turbulent diffusion coefficients and molecular diffusivities result in cumbersome transport models (e.g. Murillo et al, 2006) that may be of research utility but grants no practical advantage over simpler ones where a lump-sum mixing coefficient is used (Tsanis & Boyle, 2001). We have therefore refrained from such models. Landmark results however have been obtained by the semi-empirical determination of the mixing coefficients themselves. Elder (1959) gives the mixing coefficient as $D_x = 5.93 * u * h$ and $D_z = 0.12 * w * h$ where w is the vertical component of the velocity which we neglect in our case. Mingham (2008) suggested that the values of 5.93 and 0.12 for D_x and D_z are too low and should be closer to 12.1 and 1.2 respectively. In any case, the D 's are empirical constants that can be calibrated based on field data in our model.

The source/sink term on the RHS of equation (14) may represent pollutants entering from the boundaries or may be due to pollutant decay/growth, biological transformation, chemical reaction or a combination of these processes. Most pollutant transport models are used in estuaries, lakes, rivers and other water bodies and the source/sink term is omitted. We have included the term in our model so that it can be applied in a variety of spill scenarios. It can be set to zero if not needed in a particular situation.

2.9 Erosion and Sediment Transport Models

One of the objectives of this project is to know how much sediment is picked up and delivered to a stream or water body in an event. A list of the most used erosion-sediment transport models is given in Table 2.9.1. Following Wheater et al (1993), models are

classified as empirical, conceptual or physics-based. As Table 2.9.1 shows, this classification is not perfect as there are some hybrid models where a couple of modules would be empirical and others would be conceptual and/or physics-based.

Empirical models are generally the simplest of the three types and their data and computational requirement is usually small (Haan et al, 1994). However, they tend not to be event-responsive (Rose, 1993) since they are based on the assumption that conditions remain unchanged throughout the period under study. Being statistical models, they tend not to show causation and their parameters have no physical meaning (Jakeman & Hornberger, 1993). They are also very site-specific so that extensive calibration has to be carried out if they are to be used on a new site (Bull and Kirkby, 1997).

Conceptual models are suited for catchment scale modeling (Prosser and Rustomji, 2000). They divide a catchment into subdivisions (internal storages). The erosion processes of detachment, transport and deposition are determined both empirically and physically for each of the internal storages and the outputs are lumped (*see Bear, 1987; Sorooshian, 1991 and Spear, 1995 for details*).

Physics-based models are results of considering the physical conservation laws of mass and momentum both of the flow and sediment (Bennett, 1974). They show the cause-effect relationship between parameters that this makes them attractive as a decision-support tool. These models incorporate the sediment transport capacity formula and their output is quite sensitive to which formula is used (Persons et al, 2001; Singh& Singh, 2001). The

Table 2.9.1: Examples of Erosion/sediment transport models (Source: Merritt et al, 2003)

Model	Type^a	Scale	Input/output	Reference
Water quality AGNPS	Conceptual	Small catchment	Input requirements: High Output: runoff volume; peak rate, SS, N, P, and COD concentrations	Young et al. (1987)
ANSWERS	Physical	Small catchment	Input requirements: High Output: sediment, nutrients	Beasley et al. (1980)
CREAMS	Physical	field 40–400ha	Input requirements: High Output: erosion; deposition	Knisel (1980)
EMSS	Conceptual	Catchment	Input requirements: Low Output: runoff, sediment loads, nitrogen loads and phosphorus loads	Vertessey et al. (2001) Watson et al. (2001)
HSPF	Conceptual	Catchment	Input requirements: High Output: runoff, flow rate, sediment load, nutrient concentration	Johanson et al. (1980)
IHACRES-WQ	Empirical/ Conceptual	Catchment	Input requirements: Low Output: runoff, sediment and nutrients	Jakeman et al. (1990, 1994a,b), Dietrich et al. (1999)
IQQM	Conceptual	Catchment	Input requirements: Moderate Output: many pollutants including nutrients, sediments, dissolved oxygen, salt, algae.	DLWC (1995)
LASCAM	Conceptual	Catchment	Input requirements: High Output: runoff, sediment, salt fluxes	Viney and Sivalapan (1999)
SWRRB	Conceptual	Catchment	Input requirements: High Output: streamflow, sediment, nutrient and pesticide yields	USEPA (1994)
Erosion GUEST	Physical	Plot	Input: High Output: runoff; sediment concentration	Yu et al. (1997) Rose et al. (1997)
LISEM	Physical	Small catchment	Input: High Output: runoff; sediment yield	Takken et al. (1999) De Roo and Jetten (1999)
PERFECT	Physical	Field	Input: High Output: runoff, erosion, crop yield	Littleboy et al. (1992b)
SEDNET		Empirical/ Catchment	Input requirements: Moderate Output: suspended Conceptual sediment, relative contributions from overland flow, gully and bank erosion processes	Prosser et al. (2001c)
TOPOG	Physical	Hillslope	Input: High Output: water logging, erosion hazard, solute transport	CSIRO Land and Water, TOPOG Homepage; Gutteridge Haskins and Davey (1991)
USLE	Empirical	Hillslope	Input: High Output: erosion Wischmeier and Smith (1978)	
WEPP	Physical	Hillslope/ catchment	Input: High Output: runoff; sediment characteristics; form of sediment loss	Laflen et al. (1991)
In-stream transport MIKE-11	Physical	Catchment	Input: High Output: sediment yield, runoff	Hanley et al. (1998)

^a Model classification refers to the over-arching process representation of the model. Model components generally contain a mix of empirical, conceptual and physics-based algorithms.

sediment transport algorithms employed in popular physic-based models are described in Table 2.9.2 Physics-based model can be used effectively for small areas i.e. plot scale (Hudson, 1975) unlike empirical and conceptual models that are better suited for basin and catchment scale prediction (Loch & Silburn, 1996).

Many of the erosion-sediment transport models available have been developed by agencies who are more interested in what effect land management practices would have over time – usually years – and what best procedures to adopt (Hairsine & Rose, 1992a; Nearing et al, 1994). Their focus is mainly agricultural farmlands and the slopes considered are not very steep. A majority of them are also 1-D and the assumptions underlying their derivation e.g. uniform steady flow are not valid in our case.

Our erosion/sediment transport model is physics-based. It is an extension of Foster’s equation in 2-D. Assuming the main erosion processes take place in the rills, we have (see Appendix C for derivation):

$$\frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y} = D_f \quad (2.15)$$

where $G_x = huC_s$ and $G_y = hvC_s$ are the sediment load ($kg.s^{-1}.m^{-1}$) in the x - and y -directions respectively; D_f is the net detachment rate ($kg.s^{-1}.m^{-2}$) which is negative for sediment deposition and positive for sediment detachment and C_s is the concentration of the sediment ($kg.m^{-3}$) in the flow.

Table 2.9.2: Some commonly used algorithms for sediment transport in physics-based erosion models (Source: Merritt et al, 2003)

Name	Algorithm	Example model
Foster's equations	Steady state continuity equation for rill and interill detachment and/or deposition	WEPP NSERL (1995) CREAMS
	$\frac{dqs}{dx} = Dr + Di$	
	where $\frac{dqs}{dx}$ is the sediment rate per unit width of rill channel, Dr and Di are the rill and interill net detachment or deposition rate, respectively	
Engelund and Hansen (1968)	Fundamental energy transport equation for transport and deposition of sediments along a movable bed	TOPOG http://www.clw.csir.o.au/topog/user/
	$qT = 0.04 \frac{(Sh)^{3/2}}{(s-1)^2 d_{50} g^{1/2}} v^2 = 0.04 \left(\frac{2g}{f}\right)^{1/6} \frac{(Sq)^{5/3}}{(s-1)^2 d_{50} g^{1/2}}$	
	qT is the amount of transported sediment (m ³ m ⁻¹ s ⁻¹), S is the energy slope, s is the ratio of the specific weight or density of sediment to water, v is flow velocity (ms ⁻¹), h is water depth (m), F is the roughness coefficient, d_{50} is the median grain diameter (m), q is runoff (m ³ m ⁻¹ s ⁻¹) and g is acceleration due to gravity.	
Rose	Steady state sediment flux equation, (1) in the absence of rills, and (2) when rills are present.	GUEST Misra and Rose (1996) Ciesiolka et al. (1995) Rose et al. (1997) Hairsine and Rose (1991) Hairsine and Rose (1992a) Rose (1993)
	$\frac{dc}{dq} = \frac{(1-H)}{qQ} \left[aP + \frac{F(\Omega - \Omega_0)}{J} \right] - \frac{c}{q}$	
	c is the equilibrium sediment concentration, q is the volumetric flux of water per unit width of plot, Q is the runoff rate per unit area, P is measured rainfall rate, a is a rainfall related erodibility parameter, F is a constant relating to the fraction of the excess streamflow power effective in re-entrainment of sediments	
	$\frac{dc}{dG} = \frac{N}{GQ} \left[\{(1-Hr)Wx + Ws\} \left\{ aP + \frac{F(\Omega - \Omega_0)}{J} \right\} + a * Wu + qsli \right] - \frac{c}{G}$	
	N is the number of rills per unit width of erosion plot, G is the discharge rate, Hr is the effective surface on which a deposited layer can form, $qsli$ is a lateral sediment flux to the rill from the interill area, Wx is the rill width, Ws represents the vertical component of the wetted perimeter, and $a*Wu$ denotes the sediment contribution by rainfall detachment at a potentially maximum rate from the unshielded portion of the rill sidewall.	

2.10 Numerical Methods for Solving the Shallow Water Equations

Before the advent of computers, the St. Venant equations used to be solved analytically or graphically (Iwasa, 1988). Hydraulics research proceeded purely on the basis of theoretical assumptions refined by experimental studies (Lai, 1986). The time it takes to solve a set of shallow water equations (SWEs) of any appreciable complexity is therefore usually long and even at that the solutions obtained are very site-specific (Vreugdenhil, 1989) and not generally applicable. The only situations that yielded to analytical treatment are 1-D uniform steady flows that admit no hydraulic jumps (Liang, D. et al, 2007).

The digital age brought about the field of computational hydraulics. Hydraulic problems were solved quickly numerically and modelers were able to address a broader range of scenarios (Wu, 2008). The SWEs are partial differential equations (PDEs) that can be solved by a variety of numerical methods. The Method of Characteristics, Finite Element Method and Finite Difference Method are popular solution methods for PDEs (Strang, 1986). The SWEs are hyperbolic PDEs. An overview of classes of PDEs is given in Appendix D. A review of these numerical methods and their range of applicability is given below.

2.11 Finite Element Method (FEM)

FEM's basic idea is that a solution domain can be subdivided into smaller domains or finite elements (Rao, 1999) and so by assuming a simple form of solution for each subdomain we can approximate the solution to the entire domain (Reddy, 1993). It is used in solving boundary-value problems numerically (Cooley and Moin, 1976). Popular Computational

Fluid Dynamics (CFD) packages like FLUENT and COMSOL use the FEM as solution technique because of the range of complex scenarios they can simulate. Other well known finite-element software packages include ANSYS, ADINA and NASTRAN (Rao, 2002). The superiority of the FEM over other numerical methods shows when working in three-dimensions (Hicks and Steffler, 1995; Franca, 2002) as it can model domains with complex bathymetries, moving boundaries and irregular boundaries (Le Roux et al, 1998). Overland flow is mainly a 2-D problem so Finite Difference schemes are preferred since they are computationally less expensive and give comparable results to FEM in 2-D (Szymkiewicz, 2010).

2.12 Method of Characteristics (MOC)

The MOC was first developed as a graphical solution to PDEs by Massau in 1889 (Massau, 1900). Dronkers and Schonfeld (1955) built on this development to lay the groundwork for modern hydraulic computation. The MOC was heavily relied on (e.g. Lai, 1965a; Liggett and Woolhiser, 1967) as a way to solve the unsteady flow equations numerically at the dawn of the digital computer era. The MOC caught the attention of earlier researchers because it gives an excellent way to investigate the properties of the solution of a hyperbolic PDE since it preserves directional information and so the physical meaning of the wave-like nature of the result is apparent.

For 1-D flow, a characteristic is a path in the distance-time plane along which a certain quantity is conserved (Vreugdenhil, 1989). In 2-D space, which is the case considered here, characteristics are lines in the solution domain along which signals, or information,

propagate. In other words, information propagates through the solution domain along the characteristic curves (Ralston and Rabinowitz, 1978). Discontinuities in the derivatives of the dependent variable - if they exist – also propagate along the characteristic curves. If a PDE possesses real characteristics, then information propagates along these characteristics. If no real characteristics exist, then there are no preferred paths of information propagation. Consequently, the presence or absence of characteristics has a significant impact on the solution of a PDE (Hoffman, 1992).

The method involves transforming an original set of governing PDEs into characteristic equations. Characteristic equation is the differential equation of the characteristic curve. A suitable numerical scheme is then used to solve these characteristic equations (Abbott, 1966).

The method has some drawbacks. When convection and diffusion are both present in a physical process, information propagates by both convection and diffusion. The MOC gives large errors when handling the diffusive terms in such convection-diffusion equations (Stanoyevitch, 2005). The form of the SWEs we are solving has diffusive terms and so treating it with the MOC will yield poor results. The MOC works well with non-stiff 1-D flow problems. When flow is 2-D however or if shock waves are present, developing a MOC is difficult and in cases of severe non-linearity, the method fails (Basco, 1983). Rarefaction can also occur with this method. Rarefaction is when characteristics fail to cover part of the solution domain of the PDE.

A good understanding of the MOC is essential to the development of finite difference methods for solving hyperbolic PDEs. Physical paths of propagation of information through

the solution domain are present. Proper account of these paths of propagation must be taken in order to obtain physically correct numerical solutions of hyperbolic PDEs.

2.13 Finite Difference Method

A FD method transforms a PDE into a set of algebraic equations by discretizing the continuous physical domain, approximating the exact partial derivatives by algebraic finite difference approximations (FDAs) and substituting these FDAs into the PDE to get an algebraic finite difference equation (Burden and Faires, 2011).

FD schemes are broadly divided into two: implicit schemes and explicit schemes (Smith, 1985). An explicit FD scheme is when a variable can be computed forward in time using quantities from previous time steps. In explicit FD schemes, the new value of the variable at time $n+1$ depends *explicitly* on its value at time n . In implicit FD schemes however, the output of the time-updated variable depends – at least partially – on itself (Hoffman, 1992). Examples of explicit FD schemes are Lax-Wendroff type methods and the MacCormack method while examples of implicit FD schemes are the Crank-Nicholson method and the Preissmann schemes.

Propagation problems – like the one we are considering in this project - are initial-boundary-value problems in open domains in which the solution in the domain of interest is marched forward from the initial state, guided and modified by the boundary conditions. Propagation problems are governed by parabolic or hyperbolic partial differential

equations. A complex problem will have both. The SWEs is primarily hyperbolic by nature and secondarily parabolic due to the diffusion terms.

The major similarity between both explicit and implicit FD schemes is that both march the numerical solution forward from one time level to the next. The major difference is that the numerical signal propagation speed for explicit marching methods is finite, whereas the numerical signal propagation speed for implicit marching methods is infinite.

Explicit methods are computationally faster than implicit methods, because there are no systems of equations to solve. Thus, explicit methods might appear to be superior to implicit methods. However, the finite numerical signal propagation speed of explicit methods does not model the infinite physical signal propagation speed associated with parabolic PDEs, whereas the infinite numerical signal propagation speed of implicit methods correctly models the infinite physical signal propagation speed of parabolic PDEs. Thus, implicit methods appear to be well-suited for solving parabolic PDEs, and explicit methods might appear to be unsuitable for solving parabolic PDEs. In actuality, only an infinitesimal quantity of information propagates at the infinite physical signal propagation speed of a parabolic PDE; the bulk of the information travels at a finite physical signal propagation speed. Experience has shown that explicit methods can be employed to solve parabolic PDEs (Fennema and Chauhdry, 1990). Explicit methods match the physics more accurately, and the majority of numerical methods for solving hyperbolic PDEs are explicit methods.

Implicit FD schemes are more computationally stable than explicit ones. Implicit schemes do not place a restriction on the size of time step that can be taken and so it allows for

larger time steps. They however require a large set of algebraic equations to be solved iteratively and so can be computationally expensive (Liang et al, 2007). Explicit FD schemes are fast and well suited for the wavelike phenomena observed in hyperbolic PDEs but they can be unstable. Stability analysis (e.g. von Neumann, 1951; Courant et al, 1928) on the linearized form of the St. Venant equation shows that explicit schemes can be stable if the time step taken satisfies the Courant condition. The Courant condition is:

$$\Delta t \leq \max\left(\frac{2\Delta x}{(u + c)_{max}}, \frac{2\Delta y}{(v + c)_{max}}\right) \quad (2.16)$$

$\Delta t, \Delta x$ and Δy are the size of the time step and grid sizes respectively while

$$c = \sqrt{gh} \quad (2.17) \quad \text{is the celerity or speed of propagation.}$$

Explicit FD schemes are thus the most suitable way to solve the St. Venant equation (Esteves et al, 2000). An adaptive time step routine that increases the time step as the solution progresses and checks if the Courant condition is still satisfied can make explicit schemes even more computationally efficient.

2.14 Fractional Step Methods

Fractional step methods, proposed by Yanenko (1971), have the advantage of reducing the multi-dimensional matrix inversion problem of the SWEs into an equivalent one-dimensional problem, so the technique becomes very simple and attractive to apply (Yakimiv and Robert, 1986). They also have the advantage of solving the SWEs without the iterative steps involved in the multi-dimensional interpolation problem. No iteration is

required since only two time levels are used to advance the equations in time. The absence of iteration greatly reduces numerical diffusion and therefore makes it suitable for problems in which small time steps are taken or where small grid sizes are required (Shoucri, 2006). The linear analysis of the SWEs for the fractional step method shows the method is unconditionally stable.

We will be using the MacCormack explicit time-splitting FD scheme to solve the St. Venant equation by adapting the method described by Garcia and Kahawita (1986). The scheme has all the advantages of an explicit FD scheme and so it can handle properly the hyperbolic nature of the SWEs. The time-splitting technique – which is a fractional step method – improves accuracy of the scheme. By writing the SWEs in their conservative form, the scheme can also capture shocks (Yost and Rao, 1998).

CHAPTER 3

METHODOLOGY

3.1 Discretizing the Shallow Water Equations

In order to use the MacCormack scheme, the shallow water equations (SWEs) will be written in the conservative form i.e. in forms that readily show the mass and momentum conservations:

Conservation of Mass Equation:

$$\frac{\partial H}{\partial t} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = R - \frac{dI}{dt} \quad (3.1)$$

Conservation of momentum equation in x-direction:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = E_x \quad (3.2)$$

Conservation of momentum equation in y-direction:

$$\frac{\partial V}{\partial t} + \frac{\partial G}{\partial x} + \frac{\partial S}{\partial y} = E_y \quad (3.3)$$

Where x and y are the orthogonal horizontal co-ordinates,

$H = H(x, y, t)$ is the flow depth (m),

$U = U(x, y, t) = uh$ and $V = V(x, y, t) = vh$ are the unit width discharges in x - and y - directions respectively and u and v are the flow velocities in the x - and y - directions

respectively. Unit width discharge is the amount of fluid crossing a plane per unit width per unit time. It has units of m^2/s (see Fig. 3.1).

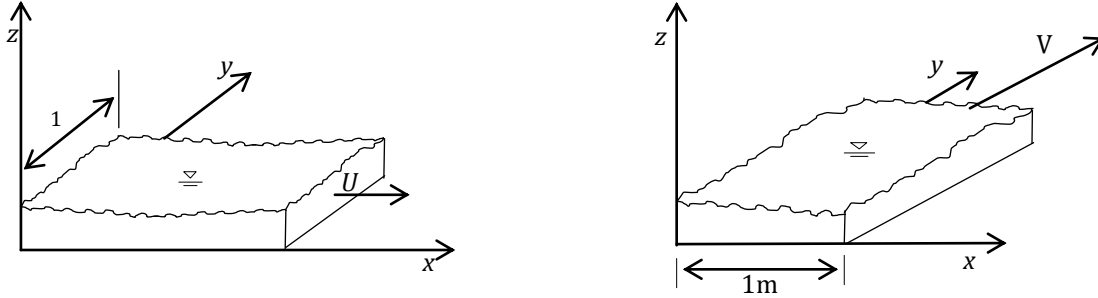


Fig. 3.1: U is the amount of fluid flowing in the x -direction per unit time per unit width of y . Its unit is $m^3/(m.s)$ or m^2/s . V is the amount of fluid flowing in the y -direction per unit time per unit width of x . Its unit is $m^3/(m.s)$ or m^2/s .

$R =$ Spill Rate (m/s)

$\frac{dl}{dt} =$ Infiltration Rate (m/s)

F , G , and S are conservation variables defined as:

$$F = F(x, y, t) = u^2 h + \frac{1}{2} g h^2 \quad (3.4)$$

$$G = G(x, y, t) = uvh \quad (3.5)$$

$$S = S(x, y, t) = v^2 h + \frac{1}{2} g h^2 \quad (3.6)$$

g is the acceleration due to gravity (ms^{-2}), $u^2 h$ and $v^2 h$ are the convective acceleration in the x - and y - directions respectively, $\frac{1}{2} g h^2$ is the gravity force term and uvh is the momentum term in a direction due to movement of fluid in an orthogonal direction.

E_x and E_y are source terms defined as:

$$E_x = E_x(x, y, t) = gH(S_{ox} - S_{fx}) + \frac{\partial}{\partial x} \left(\varepsilon_x \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial y} \left(\varepsilon_y \frac{\partial U}{\partial y} \right) \quad (3.7)$$

and

$$E_y = E_y(x, y, t) = gH(S_{oy} - S_{fy}) + \frac{\partial}{\partial x} \left(\varepsilon_x \frac{\partial V}{\partial x} \right) + \frac{\partial}{\partial y} \left(\varepsilon_y \frac{\partial V}{\partial y} \right) \quad (3.8)$$

where:

S_{ox} and S_{oy} are the bottom bed slopes defined as:

$$S_{ox} = -\frac{\partial Z}{\partial x} \quad (3.9)$$

$$S_{oy} = -\frac{\partial Z}{\partial y} \quad (3.10)$$

Z (m) is the bottom elevation from a datum, usually the sea level is used as the datum

S_{fx} and S_{fy} are the energy or friction slopes in the x - and y - directions respectively and

when approximated by the Manning formula take the form:

$$S_{fx} = \frac{n^2 U (U^2 + V^2)^{1/2}}{H^{10/3}} \quad (3.11)$$

and

$$S_{fy} = \frac{n^2 V (U^2 + V^2)^{1/2}}{H^{10/3}} \quad (3.12)$$

where n is an empirically determined constant called the Manning Coefficient

$\varepsilon_x = \varepsilon_y = \varepsilon$ is assumed, ε is the coefficient of turbulent viscosity (m^2s^{-1}) and

ρ is the density of water (kgm^{-3}) and it is assumed constant.

3.2 The MacCormack Method

The MacCormack method is very popular for solving both parabolic and hyperbolic PDEs because it can solve both linear and non-linear PDEs and a system of PDEs with equal ease. MacCormack (1969) proposed a two-step predictor-corrector FD method that uses the same grid spacing as the Lax-Wendroff (1960) one-step method. Details of the method can be found in standard texts on Numerical Methods.

3.3 The Time-Splitting Technique

Time-splitting belongs to a general class of numerical solutions methods called fractional-step methods. Yanenko (1971) developed the “method of fractional steps” which made it possible to reduce a computational solution of an initial many-dimensional problem to the consecutive solution of a number of one-dimensional problems. This turned out to be highly effective and reduces the length of time on an electronic computer by several orders of magnitude. Such a “splitting” is a reflection of the additivity of physical processes and of the spatial operators describing them. The splitting process reduces the number of calculations during each time step and achieves second order accuracy in space and time when a symmetric sequence is used.

3.4 The Numerical Grid

In order to numerically integrate the SWEs, the physical continuum region of interest is overlaid with a computational grid where all dependent variables are defined at the cell centers (fully dense grid), these values being taken to represent average cell properties.

Most previous two-dimensional models for the shallow water equations use staggered grids (e.g. Katapodes & Strelkoff, 1978; Meselhe & Holly, 1993) in which the dependent variables are defined at different points in the computational cell. Fig 3.4(a) shows an example of a typical staggered grid. Staggered grids usually generate excessive numerical diffusion compared to fully dense schemes. This artificial diffusion can create numerical circulation as well as smearing of shocks or discontinuities. Moreover, fully dense grids are conceptually more consistent than staggered grids. Fig 3.4(b) shows the fully dense grid used in the MacCormack scheme.

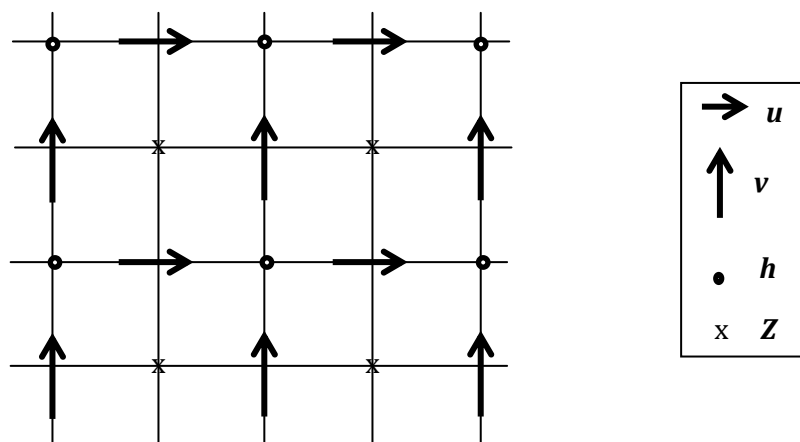


Figure 3.4(a): Typical staggered grid used in most 2-D overland flow models.

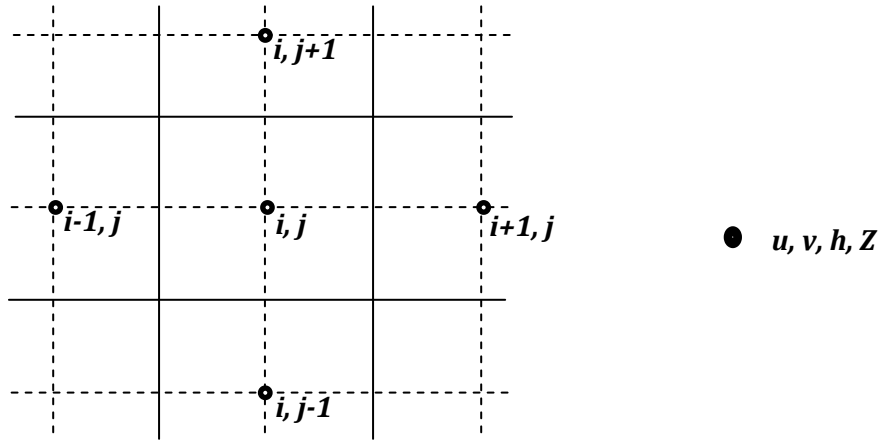


Figure 3.4(b): Fully-dense grid used in this MacCormack time-splitting scheme.

3.5 The MacCormack Time-Splitting Scheme

The MacCormack method has a slight bias in the solution due to the unidirectional nature of the backward and forward differences. This bias can be reduced by alternating the direction of the predictor and corrector differences from one time level to the next. Solution at time level $(n+1)\Delta t$, for the computational point (i, j) , is obtained through a sequence of 1-D FD operators (Γ_x, Γ_y) as follows:

$$L_{i,j}^{n+1} = \Gamma_x(\Delta t_x)\Gamma_y(\Delta t_y)\Gamma_y(\Delta t_y)\Gamma_x(\Delta t_x)L_{i,j}^n \quad (3.13)$$

$$\Delta t_x = \Delta t_y = \frac{1}{2}\Delta t \quad \text{in our model, in any case, } \Delta t_x + \Delta t_y = \Delta t \text{ .}$$

The derivatives are performed as follows:

First Γ_x operator:

Predictor: Backward differences

Corrector: Forward differences

First Γ_y operator: Predictor: Backward differences

Corrector: Forward differences

Second Γ_x operator: Predictor: Forward differences

Corrector: Backward differences

Second Γ_x operator: Predictor: Forward differences

Corrector: Backward differences

As applied to the SWEs, this procedure is:

The first Γ_x operator for the SWEs can be written as follows:

Predictor Sequence (using backward differences)

$$H_{i,j}^p = H_{i,j}^o - \frac{\Delta t_x}{\Delta x} (U_{i,j}^o - U_{i-1,j}^o) + \Delta t_x \left(R - \frac{dI}{dt} \right) \quad (3.14)$$

$$U_{i,j}^p = U_{i,j}^o - \frac{\Delta t_x}{\Delta x} (F_{i,j}^o - F_{i-1,j}^o) + g \Delta t_x \left(\frac{H_{i,j}^o + H_{i-1,j}^o}{2} \right) \left[- \left(\frac{Z_{i,j} - Z_{i-1,j}}{\Delta x} \right) - S_{fx_{i,j}}^o \right] \\ + \varepsilon_x \frac{\Delta t_x}{(\Delta x)^2} (U_{i-1,j}^o - 2U_{i,j}^o + U_{i+1,j}^o) \quad (3.15)$$

$$V_{i,j}^p = V_{i,j}^o + \frac{\Delta t_x}{\Delta x} (G_{i,j}^o - G_{i-1,j}^o) + \varepsilon_y \frac{\Delta t_x}{(\Delta x)^2} (V_{i-1,j}^o - 2V_{i,j}^o + V_{i+1,j}^o) \quad (3.16)$$

Note that R and $\frac{dI}{dt}$ are not discretized because they do not lie on the x-y plane. They are vertical quantities in the z-direction.

The “o” signifies old or previously calculated values and the “p” signifies the predicted values.

Corrector Sequence (using forward differences)

$$H_{i,j}^c = \frac{1}{2} \left[H_{i,j}^o + H_{i,j}^p - \frac{\Delta t_x}{\Delta x} (U_{i+1,j}^p - U_{i,j}^p) + \Delta t_x \left(R - \frac{dI}{dt} \right) \right] \quad (3.17)$$

$$U_{i,j}^c = \frac{1}{2} \left[U_{i,j}^o + U_{i,j}^p - \frac{\Delta t_x}{\Delta x} (F_{i+1,j}^p - F_{i,j}^p) + g \Delta t_x \left(\frac{H_{i+1,j}^p + H_{i,j}^o}{2} \right) \left[- \left(\frac{Z_{i+1,j} - Z_{i,j}}{\Delta x} \right) - S_{fx_{i,j}}^p \right] \right. \\ \left. + \varepsilon_x \frac{\Delta t_x}{(\Delta x)^2} (U_{i-1,j}^p - 2U_{i,j}^p + U_{i+1,j}^p) \right] \quad (3.18)$$

$$V_{i,j}^c = \frac{1}{2} \left[V_{i,j}^o + V_{i,j}^p - \frac{\Delta t_x}{\Delta x} (G_{i+1,j}^p - G_{i,j}^p) + \varepsilon_y \frac{\Delta t_x}{(\Delta x)^2} (V_{i-1,j}^p - 2V_{i,j}^p + V_{i+1,j}^p) \right] \quad (3.19)$$

The “c” depicts the corrected values

The first Γ_y operator for the SWE can be written as follows:

Predictor Sequence (using backward differences)

$$H_{i,j}^p = H_{i,j}^o - \frac{\Delta t_y}{\Delta y} (V_{i,j}^o - V_{i,j-1}^o) + \Delta t_y \left(R - \frac{dI}{dt} \right) \quad (3.20)$$

$$U_{i,j}^p = U_{i,j}^o - \frac{\Delta t_y}{\Delta y} (G_{i,j}^o - G_{i,j-1}^o) + \varepsilon_x \frac{\Delta t_y}{(\Delta y)^2} (U_{i,j-1}^o - 2U_{i,j}^o + U_{i,j+1}^o) \quad (3.21)$$

$$V_{i,j}^p = V_{i,j}^o - \frac{\Delta t_y}{\Delta y} (S_{i,j}^o - S_{i,j-1}^o) + g \Delta t_y \left(\frac{H_{i,j}^o + H_{i,j-1}^o}{2} \right) \left[- \left(\frac{Z_{i,j} - Z_{i,j-1}}{\Delta y} \right) - S_{fy_{i,j}}^o \right] \\ + \varepsilon_y \frac{\Delta t_y}{(\Delta y)^2} (V_{i,j-1}^o - 2V_{i,j}^o + V_{i,j+1}^o) \quad (3.22)$$

Corrector Sequence (using forward differences)

$$H_{i,j}^c = \frac{1}{2} \left[H_{i,j}^o + H_{i,j}^p - \frac{\Delta t_y}{\Delta y} (V_{i,j+1}^p - V_{i,j}^p) + \Delta t_y \left(R - \frac{dl}{dt} \right) \right] \quad (3.23)$$

$$U_{i,j}^c = \frac{1}{2} \left[U_{i,j}^o + U_{i,j}^p - \frac{\Delta t_y}{\Delta y} (G_{i,j+1}^p - G_{i,j}^p) + \varepsilon_x \frac{\Delta t_y}{(\Delta y)^2} (U_{i,j-1}^p - 2U_{i,j}^p + U_{i,j+1}^p) \right] \quad (3.24)$$

$$V_{i,j}^c = \frac{1}{2} \left[V_{i,j}^o + V_{i,j}^p - \frac{\Delta t_y}{\Delta y} (S_{i,j+1}^p - S_{i,j}^p) + g \Delta t_y \left(\frac{H_{i,j+1}^p + H_{i,j}^o}{2} \right) \left[- \left(\frac{Z_{i,j+1} - Z_{i,j}}{\Delta y} \right) - S_{fy_{i,j}}^p \right] \right. \\ \left. + \varepsilon_y \frac{\Delta t_y}{(\Delta y)^2} (V_{i,j-1}^p - 2V_{i,j}^p + V_{i,j+1}^p) \right] \quad (3.25)$$

The second Γ_y operator:

Predictor Sequence (using forward differences)

$$H_{i,j}^p = H_{i,j}^o - \frac{\Delta t_y}{\Delta y} (V_{i,j+1}^o - V_{i,j}^o) + \Delta t_y \left(R - \frac{dl}{dt} \right) \quad (3.26)$$

$$U_{i,j}^p = U_{i,j}^o - \frac{\Delta t_y}{\Delta y} (G_{i,j+1}^o - G_{i,j}^o) + \varepsilon_x \frac{\Delta t_y}{(\Delta y)^2} (U_{i,j+1}^o - 2U_{i,j}^o + U_{i,j-1}^o) \quad (3.27)$$

$$V_{i,j}^p = V_{i,j}^o - \frac{\Delta t_y}{\Delta y} (S_{i,j+1}^o - S_{i,j}^o) + g \Delta t_y \left(\frac{H_{i,j+1}^o + H_{i,j}^o}{2} \right) \left[- \left(\frac{Z_{i,j+1} - Z_{i,j}}{\Delta y} \right) - S_{fy_{i,j}}^o \right] \\ + \varepsilon_y \frac{\Delta t_y}{(\Delta y)^2} (V_{i,j+1}^o - 2V_{i,j}^o + V_{i,j-1}^o) \quad (3.28)$$

Corrector Sequence (using backward differences)

$$H_{i,j}^c = \frac{1}{2} \left[H_{i,j}^o + H_{i,j}^p - \frac{\Delta t_y}{\Delta y} (V_{i,j}^p - V_{i,j-1}^p) + \Delta t_y \left(R - \frac{dI}{dt} \right) \right] \quad (3.29)$$

$$U_{i,j}^c = \frac{1}{2} \left[U_{i,j}^o + U_{i,j}^p - \frac{\Delta t_y}{\Delta y} (G_{i,j}^p - G_{i,j-1}^p) + \varepsilon_x \frac{\Delta t_y}{(\Delta y)^2} (U_{i,j+1}^p - 2U_{i,j}^p + U_{i,j-1}^p) \right] \quad (3.30)$$

$$V_{i,j}^c = \frac{1}{2} \left[V_{i,j}^o + V_{i,j}^p - \frac{\Delta t_y}{\Delta y} (S_{i,j}^p - S_{i,j-1}^p) + g \Delta t_y \left(\frac{H_{i,j}^p + H_{i,j-1}^o}{2} \right) \left[- \left(\frac{Z_{i,j} - Z_{i,j-1}}{\Delta y} \right) - S_{fy_{i,j}}^p \right] + \varepsilon_y \frac{\Delta t_y}{(\Delta y)^2} (V_{i,j+1}^p - 2V_{i,j}^p + V_{i,j-1}^p) \right] \quad (3.31)$$

The second Γ_x Operator

Predictor Sequence (using forward differences)

$$H_{i,j}^p = H_{i,j}^o - \frac{\Delta t_x}{\Delta x} (U_{i+1,j}^o - U_{i,j}^o) + \Delta t_x \left(R - \frac{dI}{dt} \right) \quad (3.32)$$

$$U_{i,j}^p = U_{i,j}^o - \frac{\Delta t_x}{\Delta x} (F_{i+1,j}^o - F_{i,j}^o) + g \Delta t_x \left(\frac{H_{i+1,j}^o + H_{i,j}^o}{2} \right) \left[- \left(\frac{Z_{i+1,j} - Z_{i,j}}{\Delta x} \right) - S_{fx_{i,j}}^o \right] + \varepsilon_x \frac{\Delta t_x}{(\Delta x)^2} (U_{i+1,j}^o - 2U_{i,j}^o + U_{i-1,j}^o) \quad (3.33)$$

$$V_{i,j}^p = V_{i,j}^o + \frac{\Delta t_x}{\Delta x} (G_{i+1,j}^o - G_{i,j}^o) + \varepsilon_y \frac{\Delta t_x}{(\Delta x)^2} (V_{i+1,j}^o - 2V_{i,j}^o + V_{i-1,j}^o) \quad (3.34)$$

Corrector Sequence (using backward differences)

$$H_{i,j}^c = \frac{1}{2} \left[H_{i,j}^o + H_{i,j}^p - \frac{\Delta t_x}{\Delta x} (U_{i,j}^p - U_{i-1,j}^p) + \Delta t_x \left(R - \frac{dI}{dt} \right) \right] \quad (3.35)$$

$$U_{i,j}^c = \frac{1}{2} \left[U_{i,j}^o + U_{i,j}^p - \frac{\Delta t_x}{\Delta x} (F_{i,j}^p - F_{i-1,j}^p) + g \Delta t_x \left(\frac{H_{i,j}^p + H_{i-1,j}^o}{2} \right) \left[- \left(\frac{Z_{i,j} - Z_{i-1,j}}{\Delta x} \right) - S_{fx_{i,j}}^p \right] \right. \\ \left. + \varepsilon_x \frac{\Delta t_x}{(\Delta x)^2} (U_{i+1,j}^p - 2U_{i,j}^p + U_{i-1,j}^p) \right] \quad (3.36)$$

$$V_{i,j}^c = \frac{1}{2} \left[V_{i,j}^o + V_{i,j}^p - \frac{\Delta t_x}{\Delta x} (G_{i,j}^p - G_{i-1,j}^p) + \varepsilon_y \frac{\Delta t_x}{(\Delta x)^2} (V_{i+1,j}^p - 2V_{i,j}^p + V_{i-1,j}^p) \right] \quad (3.37)$$

3.6 Courant Condition

One drawback of explicit FD schemes is that they are prone to instability. An unstable solution is one that grows in an unbounded fashion. To assure stability of the scheme, the size of the time step is checked at each run to make sure it satisfies the Courant condition.

The Courant condition can be stated as

$$C_n = \frac{\text{actual wave velocity}}{\text{numerical wave velocity}} = \frac{|u| \pm c}{\Delta x / \Delta t} \quad (3.38)$$

$$\text{For stability, } C_n \leq 1 \quad (3.39)$$

C_n is the Courant number. The relation above is for the x-direction; the y-direction can be similarly stated.

The wave celerity, c , is the velocity of a small gravity wave in the flow. For a shallow-water free-surface flow:

$$c = \sqrt{gh} \quad (3.40)$$

It can be seen as the speed at which a disturbance will travel through still water. If you stand at the bank of a flowing stream of water, the velocity you will observe is, for a 1-D flow,

$$\frac{dx}{dt} = u \pm c \quad (3.41)$$

If the magnitude of the flow velocity u is greater than c , the flow is “super-critical,” if it is less than c it is “sub-critical” and if it is equal to c it is “critical.” These distinctions are important because it shows us the dominant direction of flow. For instance, for super-critical flows ($|u| > c$), the flow direction is always downstream since $u \pm c > 0$. This means super-critical flow is not influenced by any downstream feature hence no need to specify downstream boundary conditions in such a case (see Table 3.7).

3.7 Initial and Boundary Conditions for the Numerical Scheme

Hadamard (1923) states that a physical problem is well-posed if its solution: exists, is unique, and depends continuously on the boundary data and (for propagation problems) the initial data. Without a consistent set of initial and boundary conditions, no correct solution for the SWEs can be obtained. These initial and boundary conditions should be accounted for in the numerical formulation.

3.7.1 Initial Conditions

For $t=0$ the velocity field and the water depths should be specified everywhere in the numerical domain. When simulating an initially dry slope h , the flow depth, is set to a minimum value, h_{min} , rather than to zero because $h_0 = 0$ leads to numerical singularities. A value of $10^{-6}m$ is fairly common in literature as the minimum flow depth and is used here. Any cell with a lower value than h_{min} is referred to as “dry” and the flow out of such a cell is set to zero.

3.7.2 Boundary Conditions

There are several types of boundary conditions depending on the physical problem to be simulated. The two main types of boundaries we consider are Closed boundaries and Open boundaries. A problem can have both. For instance, the upstream may be an open boundary while the downstream has closed boundaries. *Geometrically* (or *geographically*) *upstream* is toward the $-x$ or $-y$ (upland) direction and *downstream* is toward the $+x$ or $+y$ (river mouth) direction; whereas, *physically*, *upstream* is the direction from which the flow is coming and *downstream* is that toward which the flow is going. Unless otherwise stated, the former convention is used throughout this project. Upstream and downstream waves correspond to the direction of the slope of the channel and not to the direction of water flow.

Closed Boundaries:

In the grid system used in this work the cell “faces” rather than the cell centers (or grid points) are aligned along solid boundaries. Therefore external fictitious points should be defined outside the numerical model (See fig 3.7.2a).

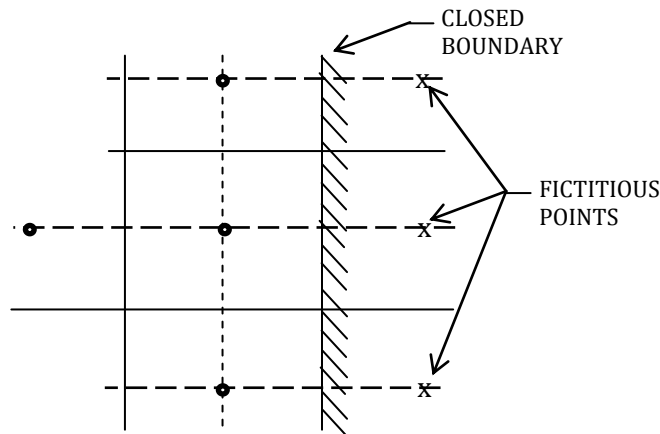


Figure 3.7.2 (a): Fictitious points along boundaries

When the coefficient of turbulent viscosity, ε is set equal to zero, slip boundary conditions should be imposed. This is accomplished using the so-called reflection technique [Insert citation] shown graphically in Fig 3.7.2(b)

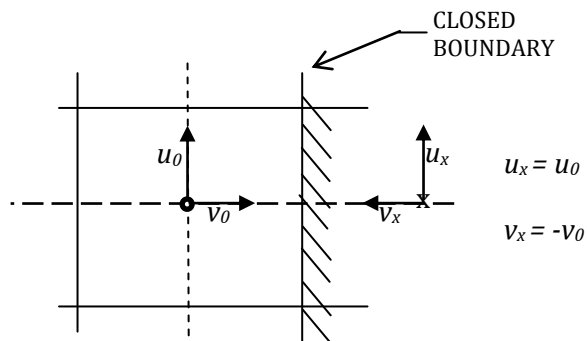


Figure 3.7.2(b): Reflection boundary condition used for slip boundaries when $\varepsilon = 0$

When ε is not zero, non-slip boundary conditions are imposed at solid “walls.” This is approximated in the manner indicated in Fig 3.7.2(c)

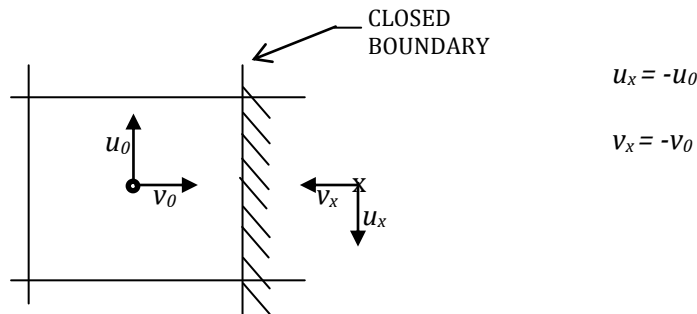


Figure 3.7.2(c): Non-slip boundary condition used when $\varepsilon \neq 0$

This approximately sets the tangential velocity equal to zero at the cell face. However, in order to correctly treat the momentum flux terms in the conservative variables (F , G , S), terms such as u^2h and v^2h are expressly equated to zero at the walls. This implies zero flux of momentum across closed boundaries.

Open Boundaries:

Flow enters and leaves the computational domain at open boundaries. Open boundary conditions are discretized in such a way as to follow as much as possible the requirements given in Table 3.7. The requirements are best seen as guidelines given by researchers (e.g. Stoker, 1957; Daubert & Graffe, 1967; Verboom et al, 1982) who have studied the mathematical basis for specifying the boundary conditions for the SWEs. For example, in a

Table 3.7: Minimum Required number of boundary conditions at open or ‘ocean’ boundaries (Source: Rene & Kawahita, 1986)

	Subcritical flow		Supercritical flow	
	Inflow b.c.s	Outflow b.c.s	Inflow b.c.s	Outflow b.c.s
1-D problems	1	1	2	0
2-D problems	2	1	3	0

two-dimensional sub-critical-flow problem for which the inflow volume flux is known, both velocity components can be specified at points located along inflow boundaries, while the water depth can be imposed at outflow boundaries.

Various types of outflow boundary conditions have been tried and will be discussed in the next Chapter on Results and Analysis. The complete listing of the programs that implement this procedure is included in the Appendix E.

3.8 Artificial Viscosity

The truncation errors associated with the FD solutions of PDEs often exhibit dispersive behavior. Pure dispersion propagates waves in space and changes the wave shape (Hoffman, 1992). At or near the sharp gradients, the dispersive errors associated with the MacCormack FD scheme produce high frequency oscillations. These oscillations have nothing to do with the physical phenomenon being simulated; they are solely due to numerical errors. By adding an explicit damping term, we may smooth these oscillations and assure good shock-capturing properties of the scheme. Jameson et al, (1981)

developed a method that can dampen these oscillations. The procedure, as adapted by Anderson et al (1984), is used in this work and it is as follows.

Compute the following parameters using the computed values of h at $n+1$ time level

$$v_{x_{i,j}} = \frac{|h_{i+1,j} - 2h_{i,j} + h_{i-1,j}|}{|h_{i+1,j}| + |2h_{i,j}| + |h_{i-1,j}|} \quad (3.42)$$

$$v_{y_{i,j}} = \frac{|h_{i,j+1} - 2h_{i,j} + h_{i,j-1}|}{|h_{i,j+1}| + |2h_{i,j}| + |h_{i,j-1}|} \quad (3.43)$$

v can be seen as the normalized form of the gradients of variable h

At the points where $h_{i-1,j}$ does not exist, use

$$v_{x_{i,j}} = \frac{|h_{i+1,j} - h_{i,j}|}{|h_{i+1,j}| + |h_{i,j}|} \quad (3.44)$$

and where $h_{i+1,j}$ does not exist, use

$$v_{x_{i,j}} = \frac{|h_{i-1,j} - h_{i,j}|}{|h_{i-1,j}| + |h_{i,j}|} \quad (3.45)$$

Similarly, at the points where $h_{i,j-1}$ does not exist, use

$$v_{y_{i,j}} = \frac{|h_{i,j+1} - h_{i,j}|}{|h_{i,j+1}| + |h_{i,j}|} \quad (3.46)$$

and where $h_{i,j+1}$ does not exist, use

$$v_{y_{i,j}} = \frac{|h_{i,j-1} - h_{i,j}|}{|h_{i,j-1}| + |h_{i,j}|} \quad (3.47)$$

then determine from the following equations

$$\epsilon_{x_{i-\frac{1}{2},j}} = \kappa \max(v_{x_{i-1,j}}, v_{x_{i,j}}) \quad (3.48)$$

$$\epsilon_{y_{i,j-\frac{1}{2}}} = \kappa \max(v_{y_{i,j-1}}, v_{y_{i,j}}) \quad (3.49)$$

where κ is a constant used to regulate the amount of dissipation. These artificial dissipative terms are used to compute the final values of the variable f at the new time step as follows:

$$\begin{aligned} f_{i,j}^{k+1} = f_{i,j}^{k+1} &+ \left[\epsilon_{x_{i+\frac{1}{2},j}} (f_{i+1,j}^{k+1} - f_{i,j}^{k+1}) - \epsilon_{x_{i-\frac{1}{2},j}} (f_{i,j}^{k+1} - f_{i-1,j}^{k+1}) \right] \\ &+ \left[\epsilon_{y_{i,j+\frac{1}{2}}} (f_{i,j+1}^{k+1} - f_{i,j}^{k+1}) - \epsilon_{y_{i,j-\frac{1}{2}}} (f_{i,j}^{k+1} - f_{i,j-1}^{k+1}) \right] \end{aligned} \quad (3.50)$$

where f is the dependent variables h , u , and v . The terms are added after a predetermined number of time steps using the latest values of h , u and v . The equation should be viewed as a MATLAB replacement statement. The procedure described above is like adding second-order dissipative terms to the SWEs. Since they are second-order, their addition does not reduce the accuracy of the MacCormack scheme. The eddy viscosity coefficient due to the

numeric of the scheme in the x -direction, for instance, is of the order of $\frac{\kappa v x \Delta x^2}{\Delta t}$.

This shows that how much the results rely on κ depends on both the gradients in the fluid depth and on the size of grid chosen. As can be seen from the equations, this κ 's influence on results in the relatively smooth regions is minimal since v tends to be zero in such a case. κ is chosen by trial-and-error: you want to select the minimum value possible that can

smoothen the wiggles near the bore. A value of κ between 0.5 and 3 is typical (Fennema & Chaudhry, 1986).

3.9 Negative Flow Depths

The SWEs are a set of constrained equations in the sense that although components of velocity can be positive or negative, the flow depth has to be positive. This physically obvious criterion is not without numerical complications. A typical contentious scenario can be seen by examining the mass conservation equation, re-written below as:

$$h^{n+1} = h^n - \Delta t \left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} \right) + \Delta t \left(R - \frac{dl}{dt} \right) \quad (3.51)$$

For the sake of argument, let us assume $R - \frac{dl}{dt} = 0$.

If $\Delta t \left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} \right) > 0$ and the $\left| \Delta t \left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} \right) \right| > h^n$, then h^{n+1} can be negative (i.e. < 0) for a very small h^n .

This problem of inadvertent negative flow depths was fixed by implementing the following procedure. If $\sum h_{adjacent\ cells} > |h_{neg}|$, the negative depth is set to zero i.e. $h_{neg} = 0$ and the flow depth of each neighbor is reduced by the product of h_{neg} and the ratio of the cell's flow depth to $\sum h_{adjacent\ cells}$. If $\sum h_{adjacent\ cells} < |h_{neg}|$, the difference is distributed among the four diagonal nodes using the same procedure. The negative cell is only set to zero when all eight of its neighbors cannot supply the amount needed to make it zero. This way, the procedure does not lead to severe mass generation.

This method is illustrated in Figure 3.9 for a case where the flow depth of computational point (i,j) is negative and water is being drawn from three of its adjacent cells.

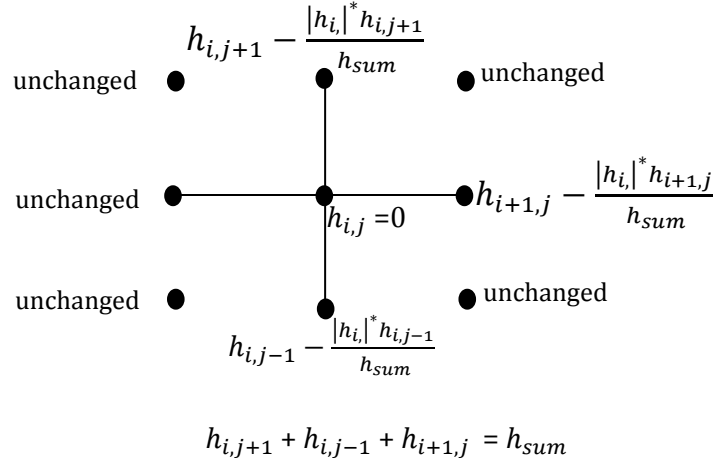


Figure 3.9: The procedure to assure a non-negative flow depth is illustrated here for a case of a cell with negative flow depth with three of its adjacent neighbors draining into it. This procedure also conserves mass. Source: Esteves et al (2000)

3.10 Bed Friction Slope

The procedure described above ensures non-negative fluid depth results after each time step but very small fluid depths can still be present. When the flow depth is very small, bed friction slope becomes very large. This process is self-aggravating: the large friction term changes the sign of the corresponding velocity component at each time step and simultaneously increases its magnitude causing some numerical instability in the solution.

This is a well known problem in the literature called the “small depth anomaly” (Tchamen & Kahawita, 1998). Finding solutions to this problem is still an active area of research. One way the situation is dealt with is to make approximations to the full St. Venant equation by

using the kinematic wave model or the diffusive wave model. As discussed in Chapter 2, we cannot use this simplistic approach in our case. Another common approach is to apply an appropriate control over the velocity in these cells e.g. setting it to zero. This however, raises the question of accuracy of the control. Some authors set the flow depth to a minimum value before momentum calculations can occur. Any small depth below this present value is “raised” to the minimum value which causes severe mass generation.

We use two original methods in dealing with these friction terms and both are explained here.

1. Avoiding division by h in the momentum equations

The reason large value of the friction slope result is because of division by h , the flow depth. If we use the Darcy-Weisbach resistance equation instead of the Manning equation, the RHS of the momentum equations changes from:

$$E_x = E_x(x, y, t) = gH(S_{ox} - S_{fx}) + \frac{\partial}{\partial x} \left(\varepsilon_x \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial y} \left(\varepsilon_y \frac{\partial U}{\partial y} \right) \quad (3.52)$$

to

$$E_x = E_x(x, y, t) = gH(S_{ox}) - \frac{fu(u^2 + v^2)^{1/2}}{8} + \frac{\partial}{\partial x} \left(\varepsilon_x \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial y} \left(\varepsilon_y \frac{\partial U}{\partial y} \right) \quad (3.53)$$

since Darcy-Weisbach formula is $S_{fx} = -\frac{fu(u^2+v^2)^{1/2}}{8gh}$ (3.54)

where u and v are the flow velocities in the x - and y - directions respectively and f is the Darcy-Weisbach coefficient. The y -direction formula can be similarly derived. The Manning coefficient n is easily available for a variety of flow scenarios. The challenge with this approach is to find reliable estimates of f . Such estimates can be known during model calibration.

2. Another way to deal with the numerical difficulty encountered in very shallow flows is to appeal to the physics of the flow. We know that the unit width discharge has to be very small (although not zero) if flow depth is very low. The procedure is as follows:

Assume the momentum equation to be solved is:

$$\frac{\partial U}{\partial t} = -gHS_{fx} \quad (3.55)$$

For shallow flows, this is a valid assumption because the friction term dominates all other terms by several orders of magnitude and so they can be safely neglected.

Discretize this equation using an implicit scheme:

$$\frac{U^{n+1} - U^n}{\Delta t} = -(gHS_{fx})^n \cdot U^{n+1} \quad (3.56)$$

Where n refers to this time step (known values) and $n+1$ refers to the next time step (unknown values). The operator splitting technique does not lend itself to implicit scheme so we re-arrange the equation as:

$$U_x^{n+1} = \frac{U_x^n}{1 + \Delta t(gHS_{fx})^n} \quad (3.57)$$

So that when S_{fx} is large, the resulting unit width discharge is small to the same degree.

The two new procedures described above do not follow strict mathematical derivations but they are physically sensible and will be shown to be reliable in practical simulations.

3.11 Infiltration Model

The dI/dt term at the RHS of the conservation of mass equation in the SWEs represent rate of infiltration of water into the soil. Only vertical infiltration is considered in this work. The infiltration rate is given by the Green-Ampt model. Groves (1989) expressed a modified form of this model as

$$\frac{dI}{dt} = \frac{K_s}{\left[1 - \exp\left(-\frac{I}{\Delta\theta\psi_f}\right)\right]} \quad (3.58)$$

where $\Delta\theta = (\theta_s - \theta)$ and ψ_f is the suction head at the wetting front, θ is soil water content, θ_s is soil water content at saturation, I is the cumulative infiltration depth and K_s is the hydraulic conductivity of the soil at saturation.

This is an Ordinary Differential Equation (ODE) within a set of PDEs. One problem encountered with incorporating the rate of infiltration is the usually high initial value of the rate. This results in negative h (flow depth) for very low initial fluid depths. This problem is solved by prescribing a minimum value of the water depth below which infiltration cannot take place:

$$h < h_{min} : \frac{dI}{dt} = 0 \quad (3.59)$$

$$h \geq h_{min} : \frac{dl}{dt} = \frac{K_s}{\left[1 - \exp\left(-\frac{l}{\Delta\theta\psi_f}\right)\right]} \quad (3.60)$$

A value of 10^{-3}m for h_{min} is common and has been used in this work.

The suction head at the wetting front, ψ_f , is calculated as:

$$\psi_f = \psi_a \left(\frac{\theta - \theta_r}{\theta_s - \theta_r}\right)^{-1/\gamma} \quad (3.61)$$

where ψ_a is the air entry suction head (m), γ is the pore-size distribution index and θ_r is the residual moisture content. Default values of the saturated hydraulic conductivity, K_s , saturated soil moisture content, θ_s , residual moisture content, θ_r , air entry suction head, ψ_a , and the pore-size distribution index, γ are given in Table 3.11 (*from Groves, 1984*). By supplying these values, the user only needs to input the value of the initial soil moisture content, θ , and the soil type to run the infiltration model. If specific field measurements of these soil properties are available, these default values can be overridden in the model. The Courant condition determines the time step to be used for the SWEs to assure stability.

However, the linear mathematical analysis that yields the Courant condition neglects the effect of source/sink terms like infiltration rate. Consequently, the time step chosen by the Courant condition for the entire SWEs may not be the best to assure the accuracy of the outputs of the infiltration model. For the Infiltration Rate routine to give accurate results, it should be able to use a time step independent of that given by the Courant condition for the full St. Venant equations. This is intuitive: being an ODE, the infiltration model needs its own time step for numerical integration. To make sure that the physics (of both the infiltration model and the SWEs) match at each computational point and at the boundaries,

the time step, Δt , of the SWEs is passed into the ODE solver written for the infiltration model. The solver then integrates the ODE from $t = 0$ to $t = \Delta t$ at each time step. Δt is seen by the solver as the run time and the solver routine can choose any time step that assures accuracy.

Table 3.11: Default Values for Soil Hydraulic Parameters. (Source: Groves, 1989)

Texture Group	Saturated Hydraulic Conductivity K_s cm/min	Saturated Soil Moisture Content θ_s cm/cm	Residual Soil Moisture Content θ_r cm/cm	Air Entry Suction Head ψ_a cm	Pore-Size Distribution Index γ
1. Loamy sand	0.1667	0.43	0.04	5.0	0.43
2. Sandy loam	0.0583	0.45	0.05	7.5	0.38
3. Loam	0.0108	0.47	0.08	13.5	0.31
4. Sandy clay loam	0.0217	0.42	0.10	7.5	0.23
5. Silty clay loam	0.0013	0.48	0.08	40.0	0.23
6. Silty clay	0.0005	0.49	0.11	65.0	0.20

3.12 Pollutant Transport Model

An advection-diffusion equation is used to show the temporal and spatial evolution of the concentration of pollutants in the flow. Guymer et al (2005) renders this relationship as:

$$\frac{\partial C}{\partial t} + \frac{\partial(uC)}{\partial x} + \frac{\partial(vC)}{\partial y} = \frac{\partial}{\partial x} \left(D_x \frac{\partial C}{\partial x} \right) + \frac{\partial}{\partial y} \left(D_y \frac{\partial C}{\partial y} \right) + \frac{source}{sink} term \quad (3.62)$$

where C is the concentration of the pollutant species, D_x and D_y are the mixing coefficients (m^2s) in the x - and y - direction respectively.

The equation is discretized the same way as the SWEs using the MacCormack time-splitting scheme. The concentration profile of a pollutant is strongly dependent on the flow velocity field. The mixing coefficients are calculated as (Elder, 1959; Mingham, 2008):

$$D_x = \alpha * u * h \quad (3.63)$$

$$D_y = \beta * v * h \quad (3.64)$$

α and β are parameters that will be determined empirically. An initial value of $\alpha = 12.1$ and $\beta = 1.2$ is suggested. This value can be refined during model calibration.

The source/sink term on the RHS of the equation represent pollutants entering from the boundaries or may be due to pollutant decay/growth, biological transformation, chemical reaction or a combination of these processes. It is generally known and supplied by the user.

3.13 Erosion-Sediment Transport Module

Solving the hydrodynamic equations give the flow depth and flow velocity fields needed in an erosion model to compute sediment discharge. Suspended sediment concentration is assumed small enough to not affect flow dynamics hence the flow velocity is taken to be the same as the sediment particle velocity in erosion modeling (Foster & Meyer, 1972; Govindaraju & Kavvas, 1991).

3.13.1 Sediment Continuity Equation

The main idea behind modeling upland erosion is that the sediment load is controlled by either the amount of available detached sediment or by the transport capacity of the flow. Bennett (1974) used the conservation of mass of sediment to derive the sediment continuity equation in 1-D. Foster (1982) modified the equation by assuming quasi-steady sediment movement:

$$\frac{dG}{dx} = D_f + D_i \quad (3.65)$$

where

G = sediment load ($kg.s^{-1}.m^{-1}$)

D_f = rill erosion rate ($kg.s^{-1}.m^{-2}$). It is positive for detachment and negative for deposition.

D_i = interrill sediment delivery to the rill ($kg.s^{-1}.m^{-2}$)

Foster's equation is used in many physically based erosion models like WEPP, NSERL and CREAMS (Merritt et al, 2003). The sediment delivery from the interrill areas is strongly dependent on rainfall intensity (Foster et al, 1981). This is because the impact of raindrops loosens the soil and the detached soils are carried off by excess rainfall. Our proposed model neglects this contribution from the interrill areas because zero raindrop impact is assumed. Tayfur (2001) justifies this approach of dropping D_i in the absence of high rainfall intensity or other processes, for example soil tillage, that may aid sediment detachment. Moreover, surface water flows are concentrated in rills and a reliable result can still be obtained by neglecting this contribution from interrill areas (Johnson & Julien, 2000; An &

Liu, 2009). Another assumption of the model is that the change in bed elevation is small so a bed update (Exner) equation is not necessary (see Abderrezzak & Paquier, 2011).

The model is:

$$\frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y} = D_f \quad (3.66)$$

The model can be seen as the Foster's equation in 2-D with the interrill sediment delivery to the rill contribution neglected. G_x and G_y are the sediment loads in the x - and y - directions respectively. A derivation from mass conservation principles can be found in Appendix C.

The eroding capacity of a flow depends on how much sediment is in it – the sediment load – compared to the maximum sediment it can hold under the same hydraulic conditions – i.e. its sediment transport capacity. If the sediment load is greater than the sediment transport capacity, deposition (of sediment) occurs; whereas if it is less, detachment (of sediment) occurs. Deposition and detachment often occur simultaneously in natural flows and so what we usually calculate is the “net deposition” and “net detachment.” The micromechanics of soil erosion is still a poorly understood phenomenon (Tang et al, 2012) and estimating net effects is standard procedure.

3.13.2 Calculation Procedure

Step 1: Compute Sediment Transport Capacity (T_c)

Of the several sediment transport formulas available, the Yalin equation (Yalin, 1963) has been found to give the most reliable estimates of transport capacity for overland flow

(Alonso et al, 1981). Finkner et al (1989) simplified the Yalin equation by identifying the key parameters that the transport capacity depends on. They reduced the equation to:

$$T_c = K_t \tau_f^{3/2} \quad (3.67)$$

where:

T_c is the sediment transport capacity ($kg.s^{-1}.m^{-1}$) and K_t is the transport coefficient ($m^{1/2}.s^2.kg^{-1/2}$) that depends on the shear stress on the soil. τ_f is the flow shear stress (Pa) acting on the soil:

$$\tau_f = \gamma h S_f \quad (3.68)$$

where:

$\gamma = \rho g$ is the specific weight of water ($kg.m^{-2}.s^{-2}$), ρ is water density ($kg.m^{-3}$) and g is the acceleration due to gravity ($m.s^{-2}$).

Step 2: Compare the magnitudes of the Sediment Load to the Sediment transport Capacity

i.e. is $G > T_c$? If Yes, Compute Net Deposition (Go to Step 4)

If No, Compute Net Detachment (Go to Step 3)

Step 3: Compute Net Detachment

Detachment occurs when the flow shear stress is greater than the critical shear stress holding the soil particles together. It is calculated as:

$$D_f = D_c \left(1 - \frac{G}{T_c}\right) \quad (3.69)$$

where:

D_f is the net detachment by rill flow ($kg.s^{-1}.m^{-2}$) and

$$D_c = K_r(\tau_f - \tau_c) \quad (3.70)$$

where:

K_r is the rill erodibility parameter ($s.m^{-1}$) and τ_c is the critical shear stress (Pa) of the soil.

No detachment occurs if $\tau_f < \tau_c$.

Step 4: Compute Net Deposition

Net deposition is computed as:

$$D_f = \alpha(T_c - G) \quad (3.71)$$

where:

D_f is the net deposition rate ($kg.s^{-1}.m^{-2}$) and α is a first-order reaction coefficient (m^{-1}). For

overland flow, Foster (1982) estimated α as:

$$\alpha = \frac{\beta V_f}{q} \quad (3.72)$$

where:

β is a turbulence coefficient, in the WEPP model, it is assigned a value of 0.5 for raindrops impacting rill flows (*after* Davis, 1978) and 1.0 for other cases like snow melting or furrow irrigation (*after* Einstein, 1968) so we take it to be 1.0 in our model; q is the unit width discharge (m^2s^{-1}) and V_f is fall velocity that can be calculated from any standard relationships, for example, the Rubey equation (Rubey, 1933).

Step 5: *Update the value of the Sediment Load, G*

Update the value of the sediment load using Eqn (3.66) with the computed value of D_f along each Cartesian coordinate.

These steps are illustrated in the flowchart shown in Fig. 3.13.

Most upland erosion schemes found in the literature are 1-D, the few available 2-D algorithms (e.g. Johnson & Julien, 2000; Tayfur 2001 and An & Liu, 2009) assume either the kinematic or diffusive wave model in solving the St. Venants equations. Their models therefore cannot handle steep slopes typically encountered in oil and gas production sites.

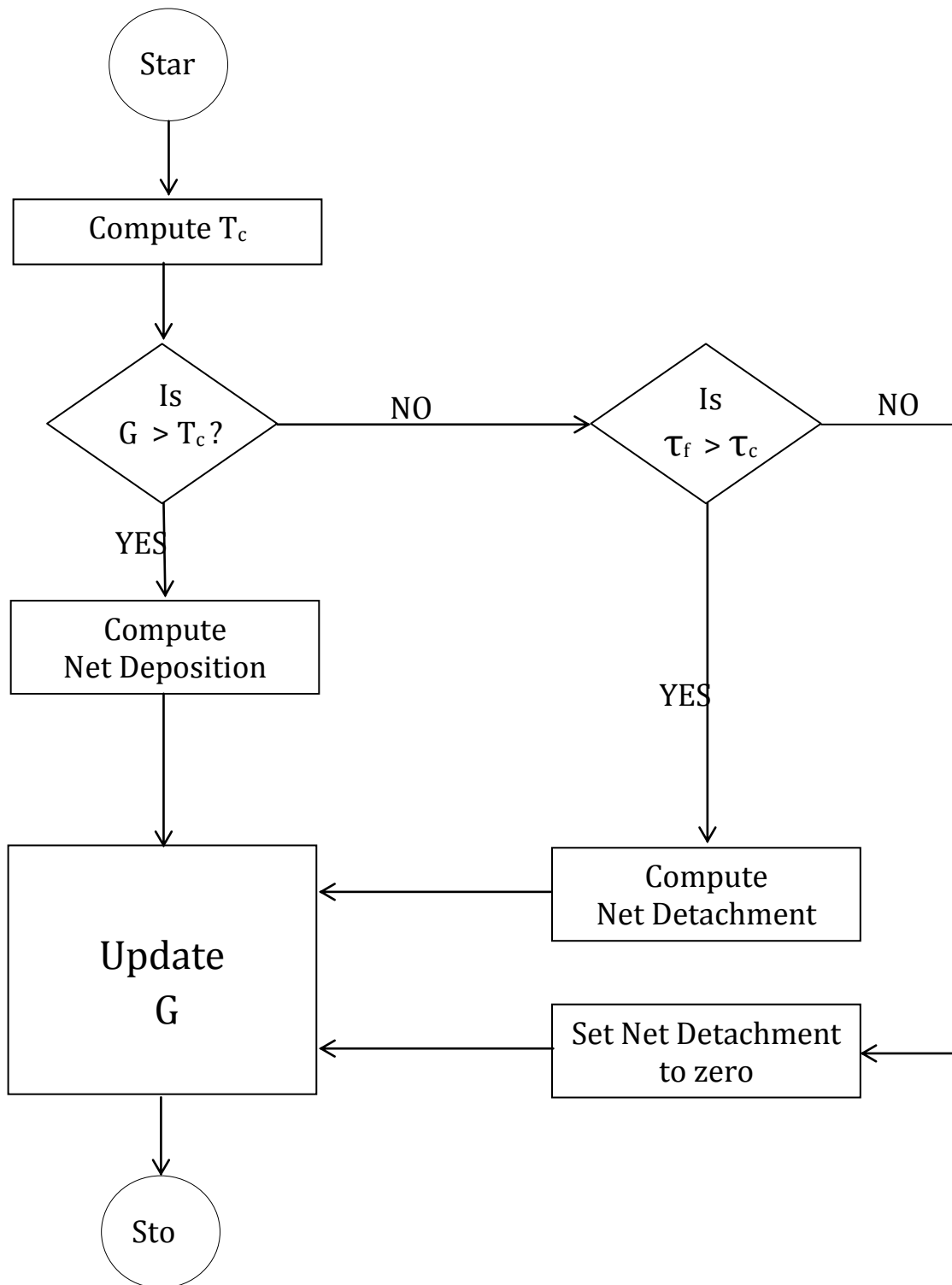


Figure 3.13: A flowchart showing how the value of the sediment load, G , is updated.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter seeks to answer the question of reliability of the model developed using the methods described in the previous chapter. For the model to be adopted in the real world, it has to perform as intended. Verification ensures that the model is programmed correctly and implemented properly and that the model captures the key features of the system being modeled. One way to verify the output of a numerical model is to compare it with analytical results. Analytical results are exact; they are the true answers to a given problem and so by seeing how a model's outputs match up with them we can assess the degree of accuracy of a numerical algorithm.

Two test problems for which the analytical solutions are known are presented in the first sections of this chapter. The performance of the model against these analytical solutions is also presented and the results are discussed. For 2D cases where the analytical solutions do not exist, Lane & Richards (2001) note that a shallow water model can be verified both numerically – for instance, by ensuring the output is independent of the size of the time step taken – and qualitatively (e.g. visualization). I relied on these two approaches in verifying the model for the more complex cases. The computer programs implementing the solutions found in this chapter can be found in the appendices.

4.2 Test Problem 1: A 1D Steady State Problem with no Shocks or Discontinuities

A channel of constant width of $10m$ and discharge $20 m^3/s$ is chosen. The channel is $1km$ long. A sketch of the problem is shown in Figure 4.2(a)

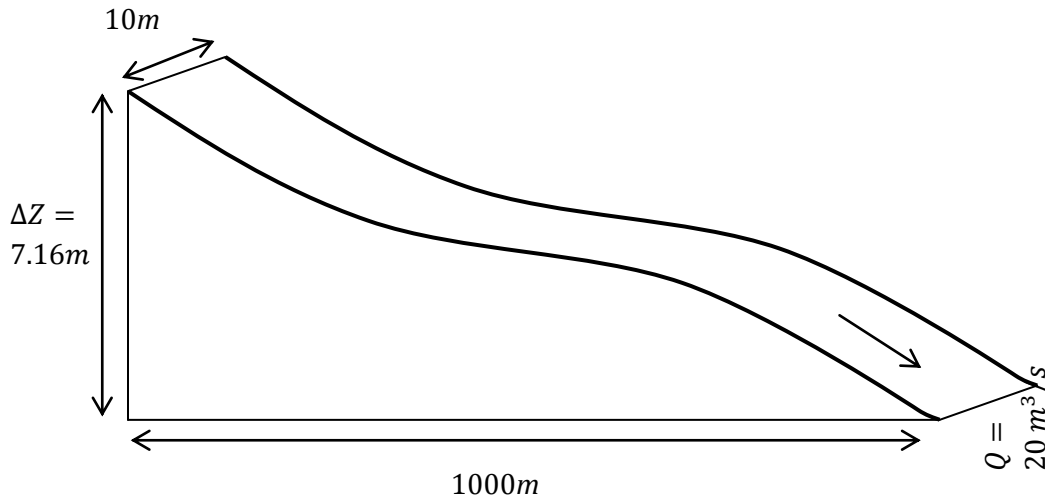


Fig. 4.2 (a): Sketch (not drawn to scale) of the 1D Steady State Problem with no Hydraulic Jump

The flow is subcritical at inflow and is subcritical at outflow with depth $0.748409m$. The Manning roughness coefficient for the channel is 0.03 . The analytical solution to this problem is given by the bed slope function:

$$S_0(x) = \left(1 - \frac{4}{g\hat{h}(x)^3}\right)\hat{h}'(x) + 0.36 \frac{(2\hat{h}(x) + 10)^{4/3}}{(10\hat{h}(x))^{10/3}}$$

where

$$\hat{h}(x) = \left(\frac{4}{g}\right)^{1/3} \left(1 + \frac{1}{2} \exp\left(-16\left(\frac{x}{1000} - \frac{1}{2}\right)^2\right)\right)$$

and

$$\hat{h}'(x) = - \left(\frac{4}{g}\right)^{1/3} \frac{2}{125} \left(\frac{x}{1000} - \frac{1}{2}\right) \exp\left(-16\left(\frac{x}{1000} - \frac{1}{2}\right)^2\right)$$

The solution to the problem is given by $h(x) = \hat{h}(x)$ where x is the distance in meters along the channel bottom. $h(x)$ is the flow depth profile and $\hat{h}(x)$ is the hypothetical flow depth profile used in deriving the analytical solution (MacDonald et al, 1995).

Figure 4.2 (b) shows the depth profile given by the analytical solution. The boundary conditions fix the depth at $0.748409m$ at both ends. The depth peaked at $1.1052m$ midway through the channel. The flow depth rises gradually from the value at the inlet boundary to this peak and falls in a similar fashion to the value at the outlet boundary. The critical flow depth value is calculated and displayed also to show that the flow is subcritical. The flow is subcritical throughout as there are no hydraulic jumps along the entire channel. Hydraulic jumps are flow control structures or natural barriers that cause flow regime to change from sub-critical to super-critical or vice versa. Hydraulic jumps introduce shocks into the solution. A shock is a mathematical discontinuity. The bed slope changes smoothly from a maximum of about 1.2% to about 0.3% midway through the channel and rises smoothly back to about 1.2%. The slope has been exaggerated a hundred times for easy display in Figure 4.2(b).

The bed level (Z), that is, the channel bottom, shown in Figure 4.2 (c) is calculated from the slope function. Since

$$S_o = -\frac{dZ}{dx} \tag{4.1}$$

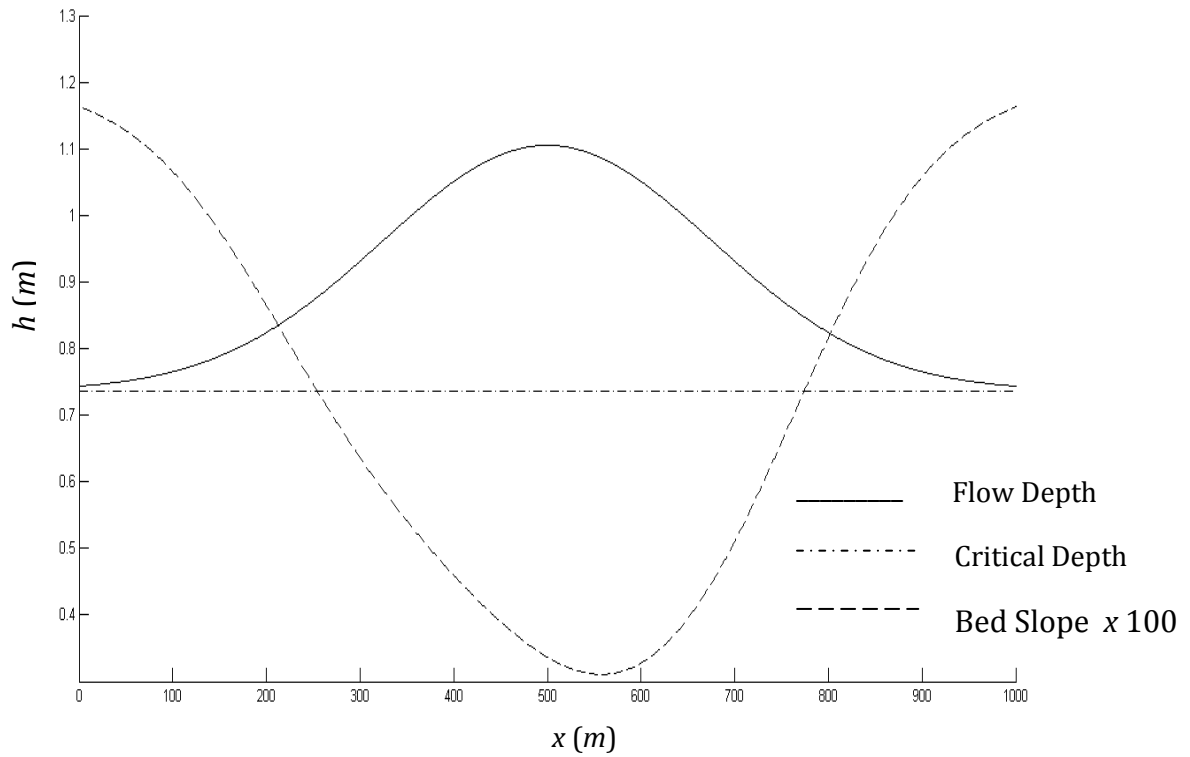


Fig. 4.2 (b): The analytical solution to Test Problem 1 is shown. The slope is magnified a hundred times for easy display.

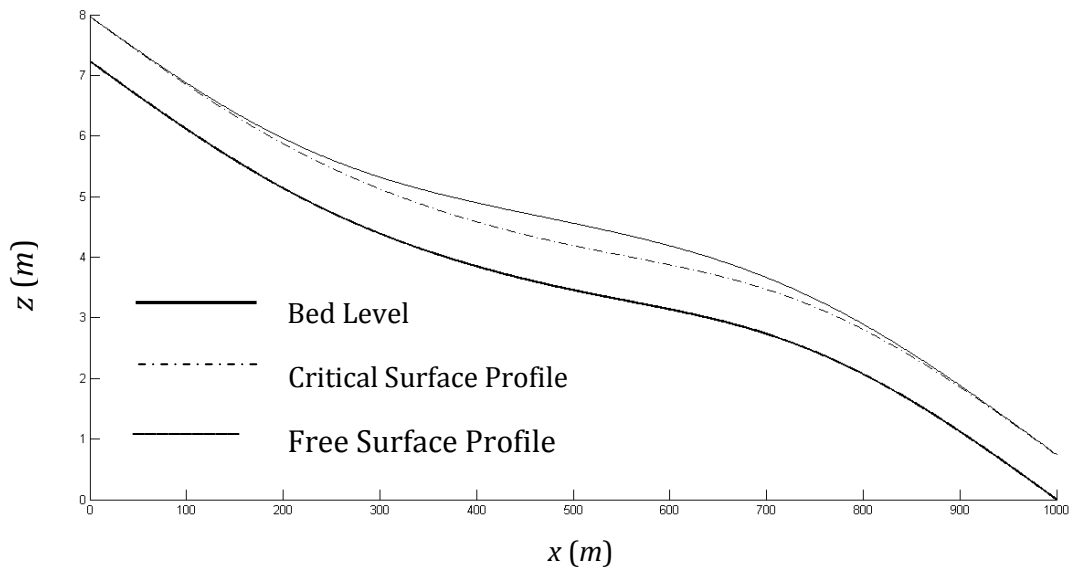


Fig. 4.2 (c): The free surface profile and bed level for Test Problem 1

$Z(x)$ is found by numerically integrating eqn. (4.1) and using a starting value such as $Z(L) = 0$. Adding the flow depth to the bed level gives the free surface profile and adding the critical depth to the bed level gives the critical surface profile. It is important to reconstruct the bed elevation, Z , from the analytical solution because it is a required input in the Shallow Water Equations (SWEs) numerical model developed here.

4.3 Numerical Solution to Test Problem 1

If a transient model like mine is given steady boundary conditions, the limiting steady state solution can be compared against the analytic steady state solution. For the problem stated in the section 4.2 above, the numerical solution is plotted and compared with the analytical solution in Figure 4.3. The numerical algorithm settings are:

$$\Delta x = 1m; \Delta t = 0.1s \text{ and } T = 800s$$

$\Delta x = \text{length of channel}/\text{number of computational points}$ is the space between one computational point and the next or the grid size. The grid size is uniform throughout the computational domain and does not change with time.

Δt is the time step used in the model. The time step is chosen to ensure that it does not impact the solution in any way. The biggest time step to give the accurate answer is 0.1s. Numerical experiments show that the result is independent of the time step used below and up to this value of 0.1s for this particular problem.

T is the time it took to reach a steady solution. Beyond 800s there is no change in the result no matter how long the simulation was left to run.

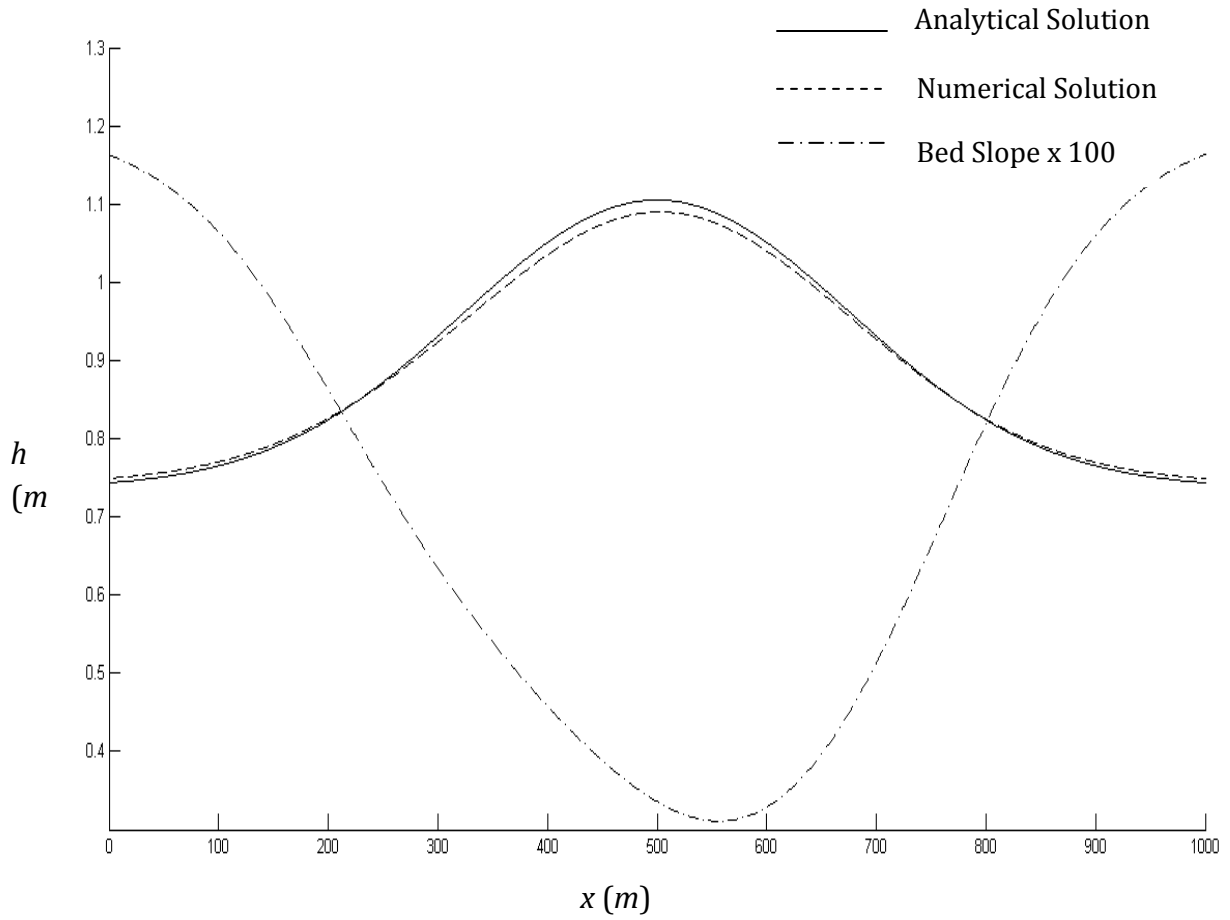


Fig. 4.3: The numerical steady solution is compared with the analytical steady solution for Test Problem 1. The bed slope is placed in the figure to show the correlation between the magnitude of the slope and the error in the numerical results.

The Courant number, C_n , for the scheme is calculated as

$$C_n = \frac{(|u| \pm c)_{max}}{\Delta x / \Delta t} \quad (4.2)$$

where c , the celerity, is \sqrt{gh} . For stability, $C_n \leq 1$.

The Courant number for the given solution is 0.55 since $\Delta t = 0.1s$.

Numerical experiments show that the scheme was unstable when $\Delta t = 0.2s$ – an outcome that is expected since $C_n > 1$ when $\Delta t = 0.2s$

When $\Delta t = 0.15s$ was used, the scheme gave a stable solution since the stability criterion is not violated ($C_n = 0.83$). Although stable, the solution given at this value of Δt was not accurate. I know this value is not accurate because I have a true answer – the analytical solution – to compare it with and also because the solution kept changing as the size of the time step used changed. This shows that stability does not ensure accuracy. A scheme may return a stable but inaccurate solution. Stability means that the error inherent in the numerical solution is not growing out of bounds but it says nothing of the accuracy of the solution itself. An accurate solution though must necessarily be a stable one. Hence, we say that stability is a necessary but insufficient condition when checking for the accuracy of a numeric scheme.

The numerical experiments conducted here to test the stability of the SWEs model has shown that the scheme employed behaved in mathematically consistent ways thus increasing the confidence we can place in its predictive ability.

The maximum relative error in the result is 1.4% and this occurs midway through the channel. The relative error was calculated as follows: $\frac{(true\ value - approximate\ value)}{true\ value} \times 100\%$

This means that even at its worst, the model was over 98% accurate for the given problem. The overall accuracy though is above 99%.

4.4 Test Problem 2: A 1D Steady State Problem with Discontinuities

4.4.1 Introduction

The previous example was important in that it shows that the model was programmed correctly, contained no bugs and that the key physics of shallow water flow were captured. The problem however is a relatively simple one because the solution was smooth; there was no discontinuity in it. Many flow problems have shocks or discontinuities because a natural or man-made barrier along the channel can cause a transition from a sub-critical flow to a super-critical one or vice versa. Critical flow occurs when Froude number is 1. The

critical depth is calculated as $h_c = \sqrt[3]{Q^2/gW^2}$ where Q is the discharge in m^3/s and W is the

channel width in m and g is the acceleration due to gravity in m/s^2 . In this section, a test problem with its analytical solution is given. In the next section, the numerical solution is compared with the analytical solution and then the role of artificial viscosity is investigated and discussed.

4.4.2 Problem Description and its Analytical Solution

The sketch of the problem is given in Figure 4.4.2 (a). The channel is still $1km$ long with a constant width of $10m$ and a constant discharge of $20m^3/s$. The flow is subcritical at inflow with depth $0.543853m$ and is supercritical at outflow with depth $1.334899m$. The cause of the sudden change in flow regime here is due to a sudden change in the bed slope. The Manning roughness coefficient for the channel is 0.02 . The analytical solution to this problem is given by the bed slope function:

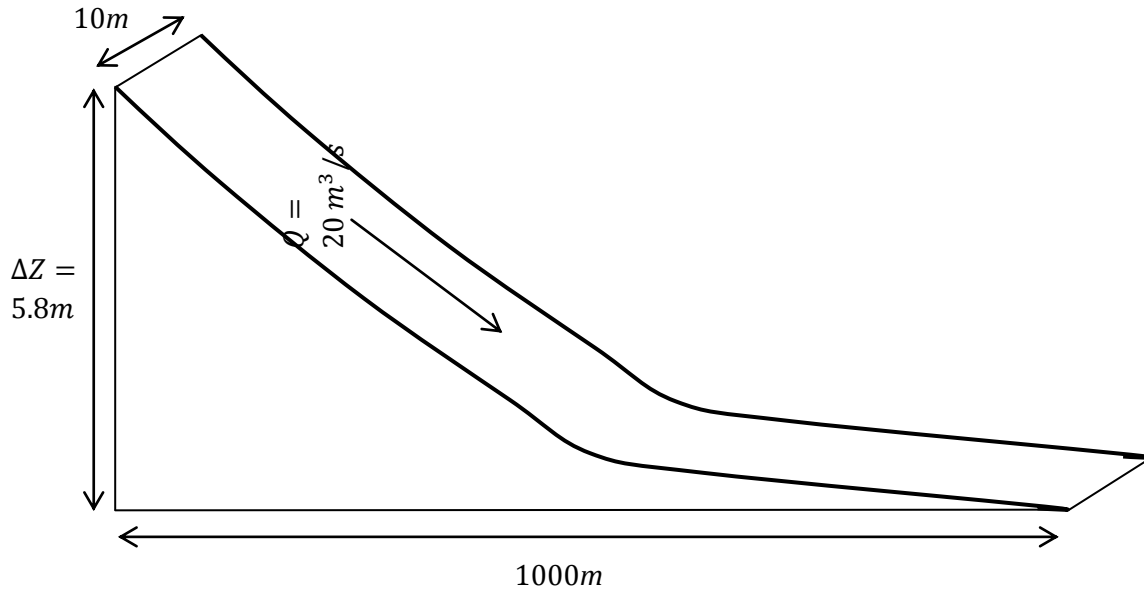


Fig. 4.4.2 (a): Sketch (not drawn to scale) of the 1D steady state problem with an hydraulic jump.

$$S_0(x) = \left(1 - \frac{4}{g\hat{h}(x)^3}\right)\hat{h}'(x) + 0.16 \frac{(2\hat{h}(x) + 10)^{4/3}}{(10\hat{h}(x))^{10/3}}$$

where

$$\hat{h}(x) = \begin{cases} \left(\frac{4}{g}\right)^{1/3} \left(\frac{9}{10} - \frac{1}{6} \exp\left(\frac{-x}{250}\right)\right) & 0 \leq x \leq 500 \\ \left(\frac{4}{g}\right)^{1/3} \left(1 + \sum_{k=1}^3 a_k \exp\left(-20k\left(\frac{x}{1000} - \frac{1}{2}\right)\right)\right) \\ \quad + \frac{4}{5} \exp\left(\frac{x}{1000} - 1\right) & 500 < x \leq 1000 \end{cases}$$

and

$$\hat{h}'(x) = \begin{cases} \left(\frac{4}{g}\right)^{1/3} \frac{1}{1500} \exp\left(\frac{-x}{250}\right) & 0 \leq x \leq 500 \\ \left(\frac{4}{g}\right)^{1/3} \left(-\frac{1}{50} \sum_{k=1}^3 k a_k \exp\left(-20k \left(\frac{x}{1000} - \frac{1}{2}\right)\right) \right. \\ \quad \left. + \frac{1}{1250} \exp\left(\frac{x}{1000} - 1\right) \right) & 500 < x \leq 1000 \end{cases}$$

With $a_1 = -0.348427$, $a_2 = 0.552264$, $a_3 = -0.555580$.

The solution to the problem is given by $h(x) = \hat{h}(x)$ where x is the distance in meters along the channel bottom. $h(x)$ is the flow depth profile and $\hat{h}(x)$ is the hypothetical flow depth profile used in deriving the analytical solution (MacDonald et al, 1995).

Figure 4.4.2 (b) shows the depth profile given by the analytical solution. The boundary conditions imposed have the depth at $0.543853m$ for the inlet boundary and $1.334899m$ for the outlet boundary. The flow depth rises smoothly from the value imposed at the inlet until halfway through the channel ($0 < x \leq 500$). There was a sudden jump caused by abrupt change of bed slope midway through the channel. The depth then rises smoothly again from this point till the end of the reach ($500 < x \leq 1000$). The critical flow depth value is calculated and displayed to show the transition from sub-critical to super-critical flow. The bed slope changes smoothly from about 1.3% at the inlet boundary to about 0.80% midway through the channel and then it changes rapidly from 0.80% to 0.35% (an over 55% change in magnitude) over a relatively short distance before it tapers off to 0.13% at the outlet boundary. Figure 4.4.2(c) shows the bed profile and the free surface profile.

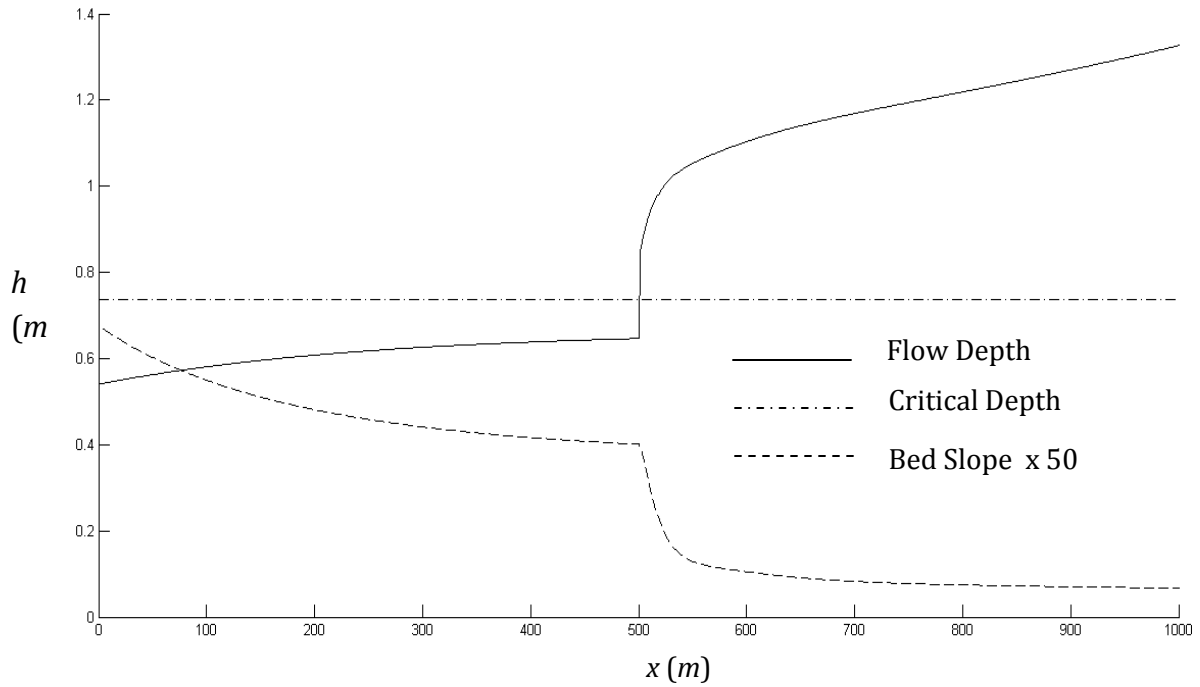


Fig. 4.4.2 (b): The analytical solution to Test Problem 2 is shown. The slope is magnified a fifty times for easy display. The abrupt change in slope caused a corresponding change in flow regime.

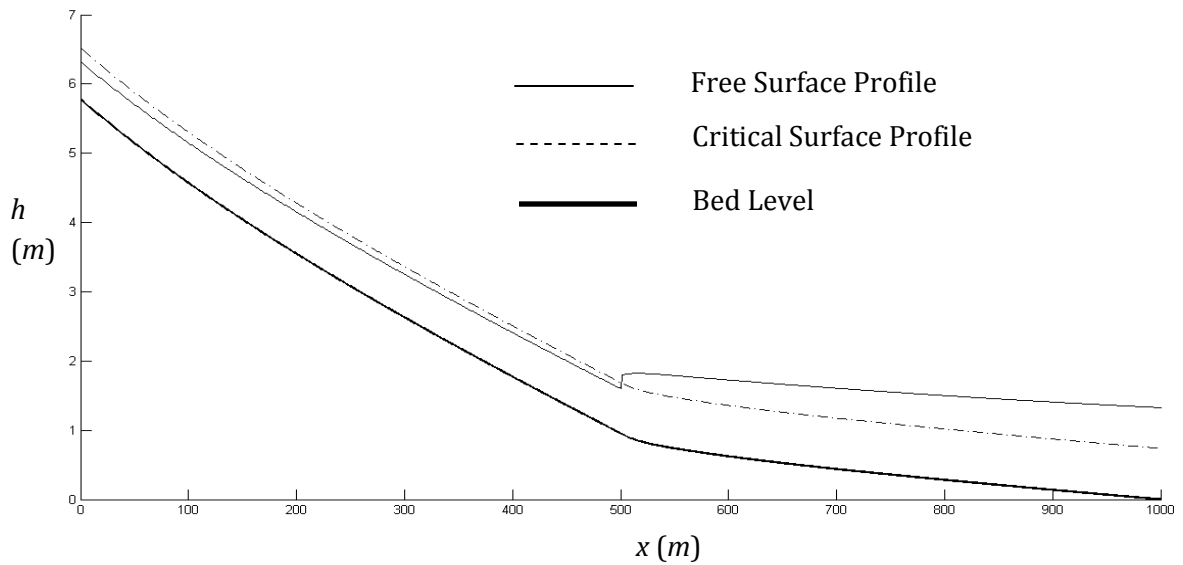


Fig. 4.4.2 (c): The free surface profile and bed level for Test Problem 2. Notice how the free surface profile became greater than the critical level midway through the channel

4.4.3 Numerical Solution to Test Problem 2

The numerical solution from the SWEs model is plotted and compared with the analytical solution in Figure 4.4.3. The numerical algorithm settings are:

$$\Delta x = 1m; \Delta t = 0.001s \text{ and } T = 2000s$$

Δx is the space between one computational point and the next or the grid size. It is constant and does not change during the simulation.

Δt is the time step used in the model. Numerical experiments show that the result is independent of the time step used below and up to this value of 0.001s for this particular problem.

T is the time it took to reach a steady solution. Beyond 2000s, numerical experiments show that there is no change in the result no matter how long the simulation was left to run.

The initial conditions are:

$$h_0 = 0.543853m \quad (0 \leq x \leq 500) \quad \text{and} \quad h_0 = 1.334899m \quad (500 < x \leq 1000)$$

$$u_0 = \frac{q}{W*0.543853} m/s \quad (0 \leq x \leq 500) \quad \text{and} \quad u_0 = \frac{q}{W*1.334899} m/s \quad (500 < x \leq 1000)$$

The relative error from between $0 \leq x \leq 500m$ is almost constant at 4.7%. Right after the hydraulic jump, the error rose to a maximum of 8.2% and then falls gradually to almost zero at the boundary. It is worth noting that even with a big abrupt change – over 55% - in the slope magnitude, the model was still over 91% accurate in the vicinity of the jump. The model results were above 95% accurate over the entire domain.

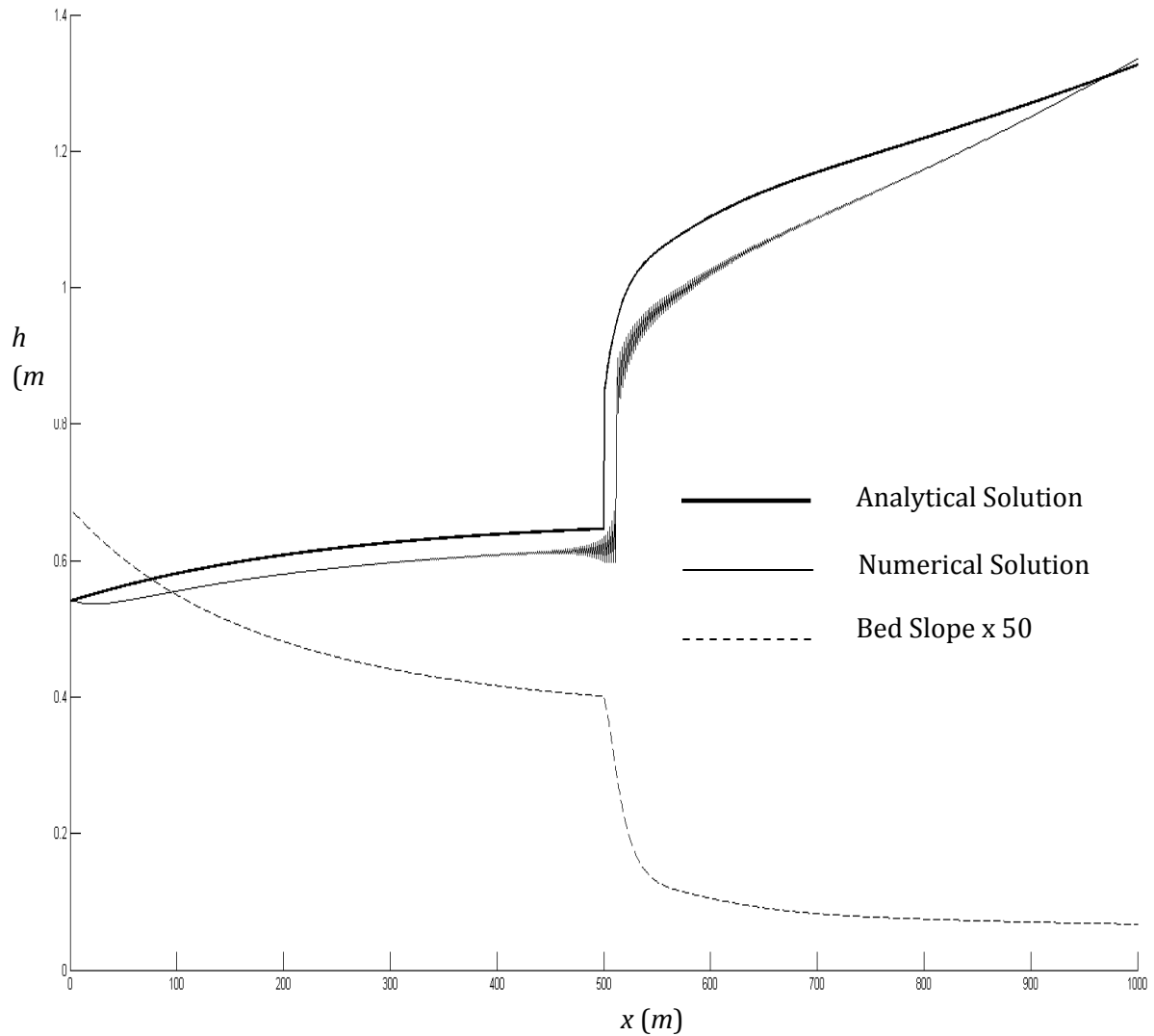


Fig. 4.4.3: The numerical steady solution is compared with the analytical steady solution for Test Problem 2. High frequency oscillations are seen at the points of abrupt changes in the depth profile. The bed slope is placed in the figure to show the correlation between the magnitude of the slope and the error in the numerical results.

This shows the ability of the model to capture shocks and its ability to handle supercritical and subcritical flows simultaneously. Although the model was reasonably accurate for this difficult problem, the result was plagued with high frequency oscillations that were purely

due to the numeric of the MacCormack scheme. At transitions points in the depth profile, these oscillations can be clearly seen producing the thick black regions observed in Fig. 4.4.3. These spurious oscillations can be observed along the entire depth profile although it is more pronounced at points where the flow regime changes.

4.4.4 Artificial Viscosity applied to the Numerical Solution to Test Problem 2

“Artificial viscosity” is a term first used by von Neumann & Richtmyer (1950) to allow for calculation of shock waves in inviscid gas dynamic equations. It is a term explicitly added to the governing equations to correct for the dispersive errors that arise due to truncation of the Finite Difference Equations (FDEs). It is needed especially in problems with shocks to smoothen out the high frequency oscillations usually observed in the solution. Roache (1972) noted that most popular numerical methods touted as having no artificial viscosity actually do have one when applied to steady-state problems.

The way artificial viscosity was calculated in this project was detailed in Chapter 3. The dependent variables are updated as follows:

$$h = h + \kappa\Lambda(h) \tag{4.3}$$

$$u = u + \kappa\Lambda(u) \tag{4.4}$$

where κ is the constant used to regulate the amount of dissipation and Λ is the artificial viscosity operator. Figure 4.4.4 shows the result when artificial viscosity is applied to the numerical algorithm for solving Test Problem 2. A value of $\kappa = 0.1$ was used in the simulation. The lowest value of κ to give a smooth solution is chosen for a given simulation. When κ is below 0.1, some oscillations were still seen in the numerical solution.

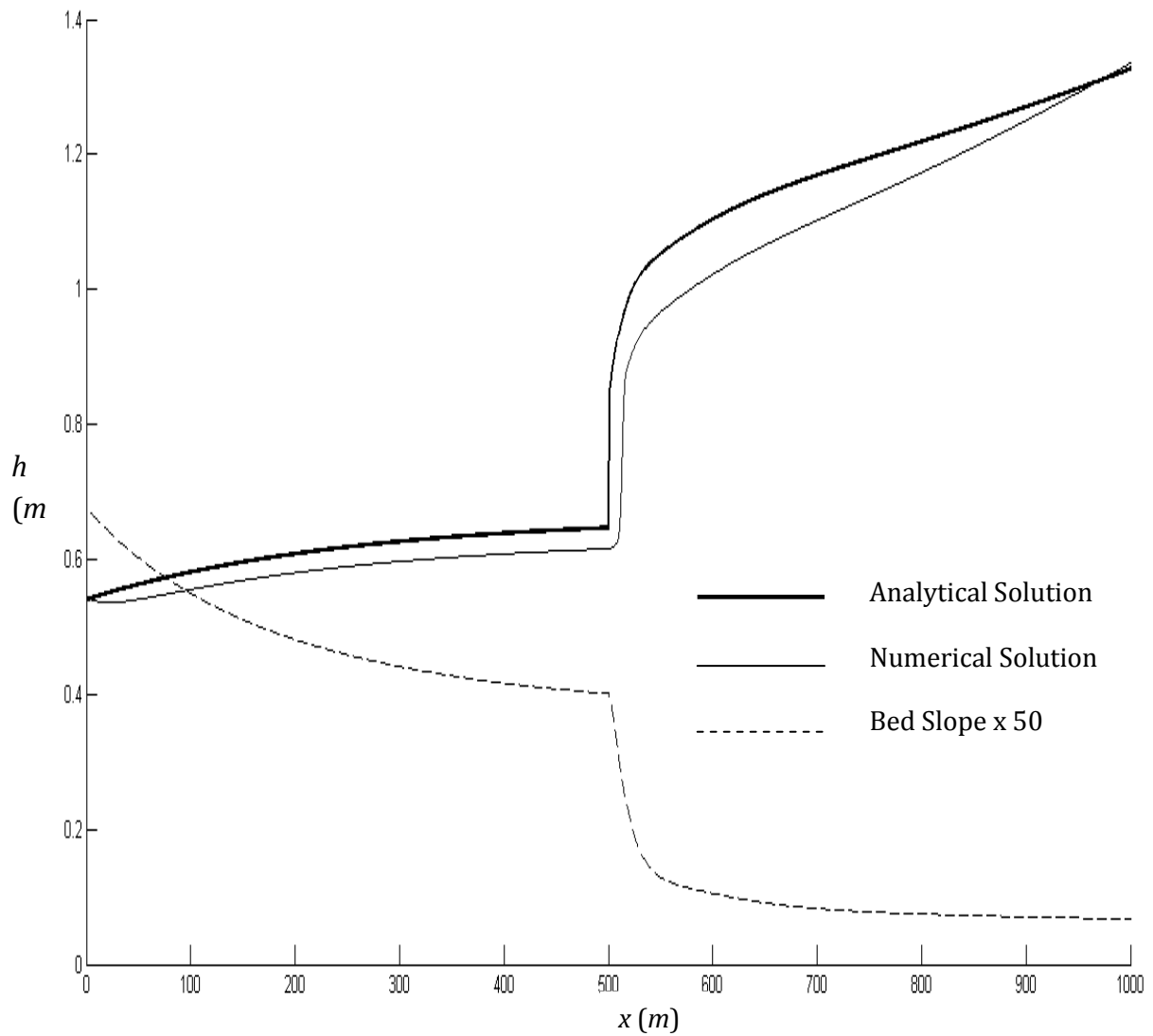


Fig. 4.4.4: The numerical steady solution with artificial viscosity is compared with the analytical steady solution for Test Problem 2. The high frequency oscillations seen in Fig. 4.4.3 are virtually gone here showing both the usefulness and efficacy of the artificial viscosity algorithm. The bed slope is placed in the figure to show the correlation between the magnitude of the slope and the error in the numerical results.

4.4.4.1 Choice of κ when using Artificial Viscosity in a Simulation

As has just been shown artificial viscosity can be very useful in smoothing out all the oscillations that arise in a problem with shocks. It should be kept as small as possible by choosing the minimum possible κ that can smear out the fluctuations. By adding artificial viscosity to the original governing equations, we are modifying the physical process so that it can be more easily computed (Guermond et al, 2011). This is why it has to be kept to the barest minimum because a large value of κ can fundamentally change the numerical method (Kurganov & Lin, 2007).

How does one know which value of κ to use for a given problem? The following is suggested:

1. Run the simulation without the artificial viscosity term (i.e. with $\kappa = 0$). If you get a smooth solution like I did in Test Problem 1 above, then artificial viscosity is not needed.
2. If there are some oscillations in the solution like in Test Problem 2, start with a very low value of κ and test to see if the oscillations will go away. If oscillations remain, start increasing κ gradually till you arrive at the minimum value needed to smoothen out the high frequency oscillations. I suggest a starting value of $\kappa = 0.001$. Values as high as $\kappa = 5$ has been reported in the literature (for instance, Hartmann & Houston, 2002). The danger in using large values of artificial viscosity is that it can hide serious errors in your numerical algorithm: the oscillations are due to instability and the instability may be due to an error in the computer program.

Artificial viscosity should not be used to gloss over flaws in the algorithm (Shu, 2009). I will be wary of using any value of κ greater than 1.

3. In some difficult flow problems, the program would not run with $\kappa = 0$. In these instances, choose the lowest possible value of κ that would get the simulation going.

I ran the simulation of Test Problem 2 with increasing values of κ to see how it affects the result. The time it took to reach the steady state solution decreased with increasing value of κ . This is to be expected because larger κ values meant the oscillations were smoothing out faster and hence the solution converged much earlier than with lower κ values. The highest value of κ to return a solution is 3.7. For $\kappa > 3.7$, the program became too unstable to run. This shows that at its best, “artificial viscosity trades accuracy for stability as the final solution is a smeared-out version of the true solution; at its worst, increasing artificial viscosity completely destroys the accuracy or even destabilizes the calculation” (Laney, 2001).

4.5 Discussion of Model Performance based on the two Test Problems

Taken together, the two test problems for which known analytic solutions exist has helped in accessing the performance of the model. I have shown that for relatively easy flow problems, the accuracy of the model is over 99%. The model has also been shown to handle discontinuities reasonably accurately. For the problem with hydraulic jump, the accuracy of the results was over 91% at the vicinity of the jump and more than 95% over the entire domain. The importance and efficacy of artificial viscosity was also seen as it

helped in smearing shocks introduced by the rapid change of slope encountered in Test Problem 2. Lastly, some guidelines on the choice of κ was given.

In the following sections, 2D cases will be investigated. There are no known analytical solutions for any 2D flow problem (MacDonald et al, 1995). But since I have shown that the scheme employed gives acceptable performance over some test 1D problems, even though there may be no theory guaranteeing good results for the more practical 2D problems, we can have some confidence in the method.

4.6 Introduction to 2D problems

The general behavior of the model was tested with different numerical experiments. The first class of experiments was carried on a frictionless flat surface so that fundamental characteristics of the scheme could be demonstrated. A real terrain from an oil and gas drilling site was then used to assess the performance of the model and investigate its adaptability to real-life scenarios.

4.7 Influence of Boundary Conditions

We examine the effects of boundary conditions on the behavior of the set of coupled equations. Boundaries are important because they continuously affect the nature of the solution (Hadamard, 1923). We demonstrate the effects of boundary conditions on the model in three different ways. The number of boundary conditions required depends on whether the flow is sub-critical or super-critical and on whether the boundary is closed i.e. reflective or open i.e. freely transmitting (Verboom et al, 1982). For all the cases shown in

Figure 4.7, we assumed a frictionless flat impermeable surface ($S_o = S_f = dl/dt = 0$), and a constant spill rate of $2 \times 10^{-5} \text{ m/s}$ at the centre of the computation domain. The simulated time was 15 mins , the time step was 3 s and the grid size was $\Delta x = \Delta y = 5 \text{ m}$.

The initial conditions are: $u(x,y,0) = 0$; $v(x,y,0) = 0$; $h(x,y,0) = 0.1 \text{ m}$.

An initial flow depth of 0.1 m may seem unreasonable at first glance but it is much lower than what other works use in validating their models. The premier Hydro-environmental Research Centre at Cardiff University, UK use an initial water level of 1.0 m when solving the St. Venant's equations numerically (*see, for example*, Liang et al, 2007, 2010) and the most cited paper – Garcia and Kahawita (1986) – on solving the Shallow Water Equations (SWEs) using the MacCormack scheme started with an initial water level of 2.0 m in one numerical experiment and a 10.0 m initial level in another. The reason for these arbitrarily high starting water levels is not far-fetched. Explicit FD schemes are the most suited for solving hyperbolic PDEs like the SWEs because they closely match the physics of the situation but explicit schemes are prone to instability. To assure stability, the Courant number has to be less than 1. The Courant number is calculated as:

$$C_n = \frac{\text{actual wave velocity}}{\text{numerical wave velocity}} = \frac{|u| \pm \sqrt{gh}}{\Delta x / \Delta t} \quad (4.5)$$

The closer C_n is to 1, the nearer it is to becoming unstable. So there is an inherent trade-off in all explicit schemes between robustness and computational speed. You cannot start with a very low value of h and have a fast numerical solver or vice versa without compromising stability.

MATLAB was used in displaying the results. The flow depths are color coded. This means that cells with the same flow depth have the same color. They can be viewed as contour lines in a physical map.

In Figure 4.7(a), a reflecting boundary was imposed on all sides of the system. This is the same as having solid “walls” on both the upstream and downstream boundaries. This means that flow was not allowed out of the computational domain and is thus “reflected” back. Calculations showed that mass was conserved as no mass was lost through the boundaries. Figure 4.7(b) depicts a non-reflecting boundary condition at both upstream and downstream boundaries. Here, mass (or any disturbance) generated within the computational domain can leave. This is also called an “ocean” or “freely transmitting” boundary. Numerical tests showed that there was loss of mass. Figure 4.7(c) is a hybrid of reflecting and freely transmitting boundary condition. The upstream is closed while the downstream is open.

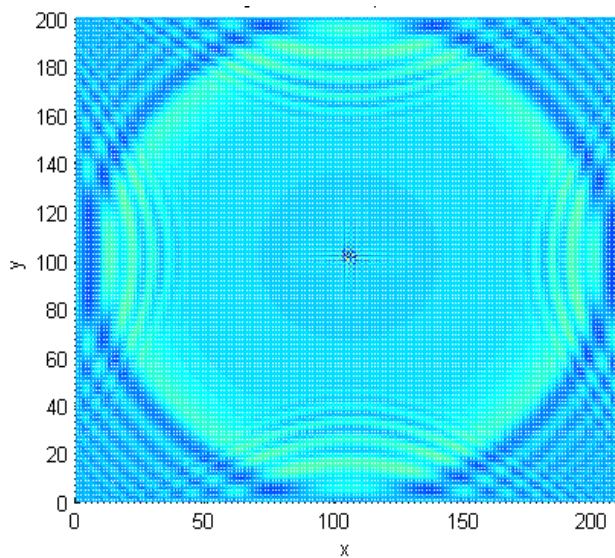


Fig. 4.7(a)

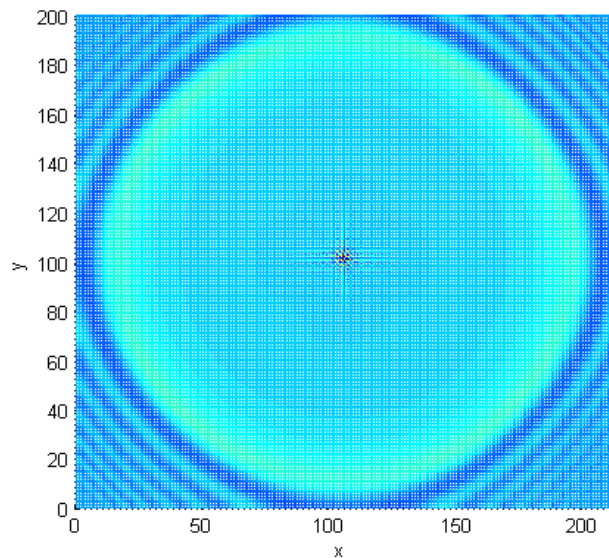


Fig. 4.7(b)

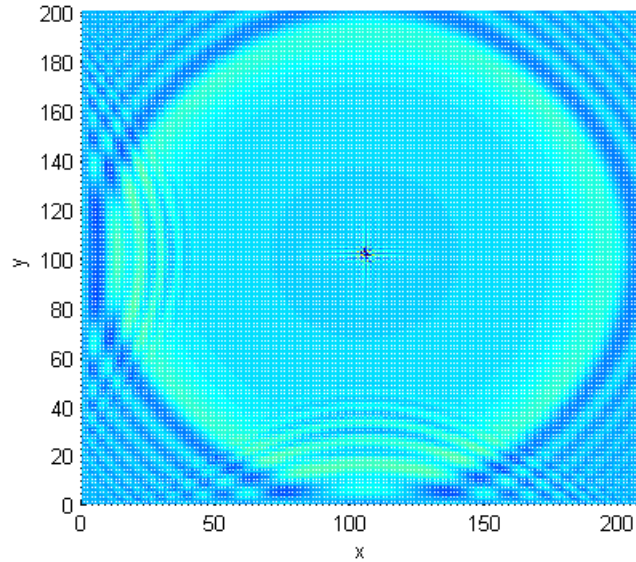


Fig. 4.7(c)

Figure 4.7: The influence of boundary conditions is seen affecting the nature of the solution continuously throughout the domain as a well-posed problem with an equally efficient solution should. *(a)*: the closed boundaries reflect waves and flow is not allowed out of the computational domain giving the flow pattern seen *(b)*: all four boundaries are open i.e. transmit waves and therefore no reflection can be observed *(c)* is a hybrid of *(a)* and *(b)* in which the upstream is closed and the downstream is open.

In all three cases, flow starts from the center of the computational domain and proceeds radially outward until it reaches the boundary where it is either reflected back or propagated out. This result shows that the SWEs are a well-posed problem with unique non-trivial solutions.

4.8 Sub-Critical Flow Problem

The problem here is similar to the freely transmitting boundary case in the previous section with one important difference: the spill source was turned off after one minute. The behavior of the propagating wave was then observed to test the ability of the model to

handle sub-critical flows. The problem is subcritical because c , the wave celerity, is greater than the flow velocity: $c = \sqrt{gh} \approx 1 \text{ m/s}$ whereas the velocities both in the x and y -directions are of the order 10^{-5} m/s . The velocities are justifiably small because in SWEs, flows are primarily gravity driven and since the surface is flat, there is no gravity force due to bottom slope. The actual movement of fluid taking place is due to a combination of convective acceleration (u^2h or v^2h) and pressure force ($1/2 gh^2$) and these are miniscule for a fluid initially at rest. Figure 4.8 (a) shows the flow pattern of the simulation by displaying the quiver plot of the velocities. The computational domain size is (208,208) which means there are over 40,000 cells in it. Arrows representing the magnitude of the velocity of each cell are thus too small to be appreciated. To correct this problem, we developed a routine that reduced the size from (208,208) to (26, 26) in which each new pixel is the arithmetic average of eight neighboring pixels in the old array. A general procedure of how to do this is in Appendix F.

The length of each arrow in Fig. 4.8 (a) indicates the magnitude of the velocity. The greater the magnitude of the velocity, the longer the arrow. The arrow head shows the direction of flow. It can be seen that disturbance has virtually left the center of the computational domain and it is spreading outwards. The wave front has not reached the distant corners of the domain because it moves in a circular fashion. The wave is moving in both $\pm x$ - and $\pm y$ -direction as it should be because the velocity seen by a stationary observer is u or $v \pm c$. At the crest of the wave front the velocity is zero and velocities of unequal magnitude in opposite directions are seen at both sides of it. This is not unlike what is observed when a wave passes through a string. This problem shows why the MacCormack method is

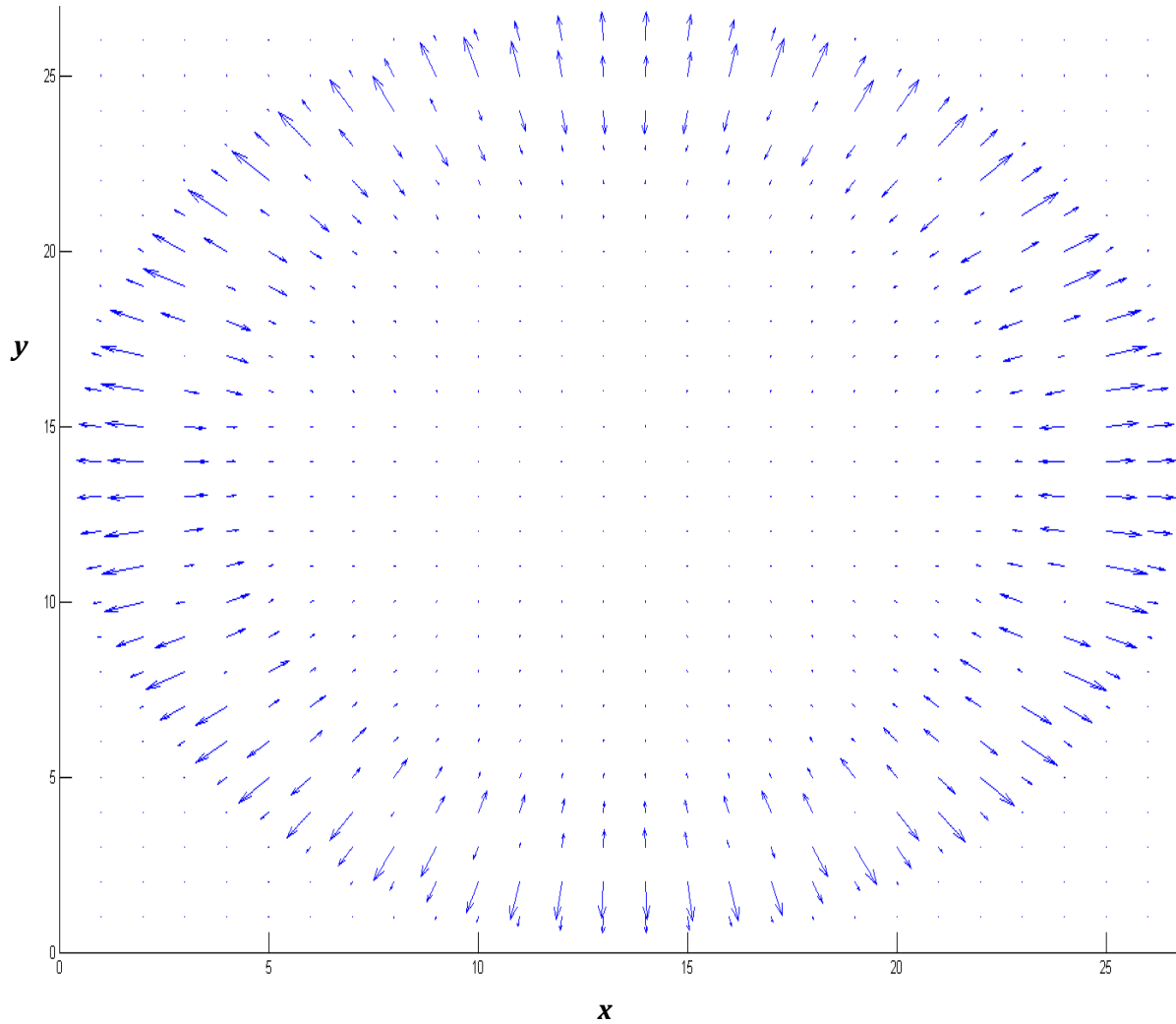


Figure 4.8 (a): Quiver plot of a flow whose spill source was turned off after a minute. The flow field shows that the disturbance has left the source (center of domain) and is heading out demonstrating the ability of the scheme to handle sub-critical flows.

preferred when solving the SWEs because few other schemes can show the direction of propagation as explicitly as displayed in the figure (Vreughendil, 1989). Fig. 4.8 (b) shows the surface profile of the flow depth. This aids in the visualization of the result just discussed. The wave front raises the flow depth and is differently colored than the rest of

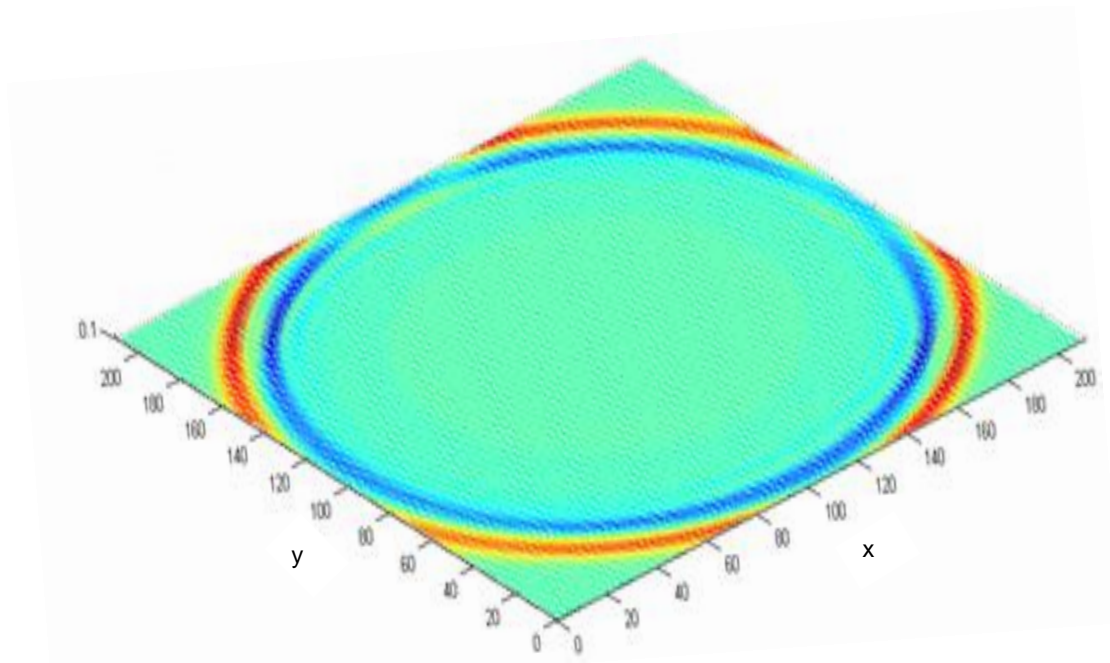


Fig. 4.8(b): Flow wave pattern after 15mins. The spill was initially generated at the center of the domain and continued for 1min. Waves can be seen propagating out of the system

the domain. The size of the domain in Fig. 4.8(b) has not been reduced like in Fig. 4.8 (a) since there is no need to do so.

A pollutant model was coupled with the SWEs. The pollutant concentration was 20 parts per million (ppm) and it also continued for a minute before it was shut off. The location of the source of the pollutant is at the centre of the domain. Fig. 4.8(c) shows the surface profile of the concentration. Data from this experiment shows that no pollutant species was seen elsewhere in the domain apart from the cell where the source took place. This is reasonable given that the actual velocity of the flow (as opposed to the celerity of the wave passing through the flow) is of the order $10^{-5} m/s$ so that in 15mins, the actual displacement is still much less than a meter. It can be argued that the velocity of the flow is not the only

factor affecting the movement of pollutant species. The mixing coefficients D_x and D_y are also important in this regard. However, the mixing coefficients are themselves strongly dependent on the velocity ($D_x = \alpha uh$ and $D_y = \beta vh$, where α and β are constants) so that a low value of velocity will give a correspondingly small value of mixing coefficient. The size each cell is $5m$ by $5m$ so it is not surprising that all of the pollutants released are still within the vicinity of their origin.

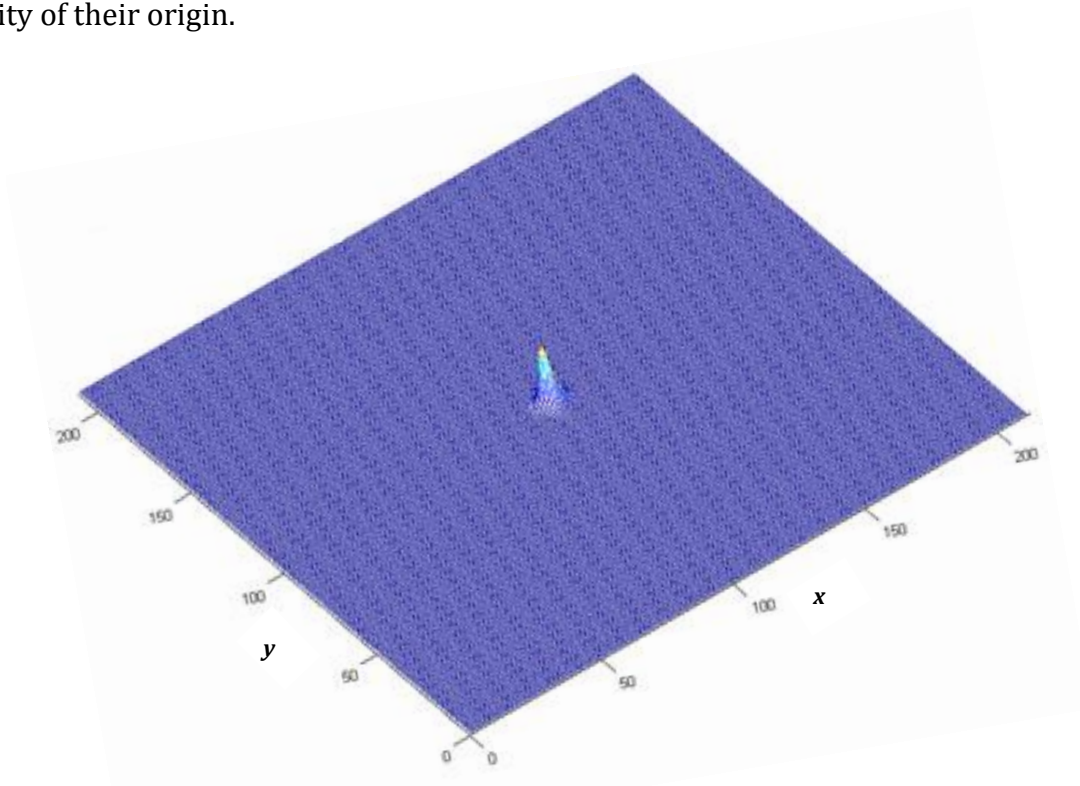


Fig. 4.8(c): The surface profile of the pollutant concentration at the end of the $15min$ simulation. All the released pollutants are still at the origin cell as physically expected. This shows the pollutant model was successfully coupled with the rest of the SWEs

By accurately handling sub-critical flows and pollutant transport in sub-critical flows, our confidence in the predictive ability of the model is increased.

4.9 Flow Over Infiltrating Surfaces

The cases considered so far involved flow over an impermeable surface in which no infiltration occurs. We now turn our attention to the problem of flow over an infiltrating surface. The domain dimensions and boundary conditions are the same as before. Sand, loam and clay are the three different soil types considered in this simulation. The values of the soil properties such as pore-size distribution index, air entry suction head, moisture content at saturation and residual moisture content were taken from Groves (1984).

The same value of initial moisture content was used in all three cases. The simulated time was $1min$ and a constant spill of $4 \times 10^{-5} m/s$ throughout the simulation was assumed. The added complexity of an infiltration model to the SWEs makes the numerical scheme less robust. To assure a stable solution therefore, the time step was reduced from $1s$ that was used in the last section to $0.05s$. The rate of infiltration was set to zero if the fluid depth or the velocity is below a certain minimum value to speed up the computation and to improve accuracy. The results are as shown in Figs. 4.9 and compared in Table 4.9. To obtain these plots, the initial flow depth was subtracted from the final flow depth so that the vertical infiltration into the soil can be readily seen. An initial infiltration depth of $0.01mm$ was used for the three soil type.

Fig. 4.9 (a) is the result obtained for flow over sand, fig. 4.9 (b) is for flow over loam and fig. 4.9 (c) is for flow over clay. Since the initial moisture content and starting infiltration depth are the same in the three situations, the rate of infiltration became strongly dependent on the pore-size distribution index. The pore-size distribution index is an empirically derived

constant that relates the pore-size distribution of a soil to its ability to retain water (Assouline, 2005).

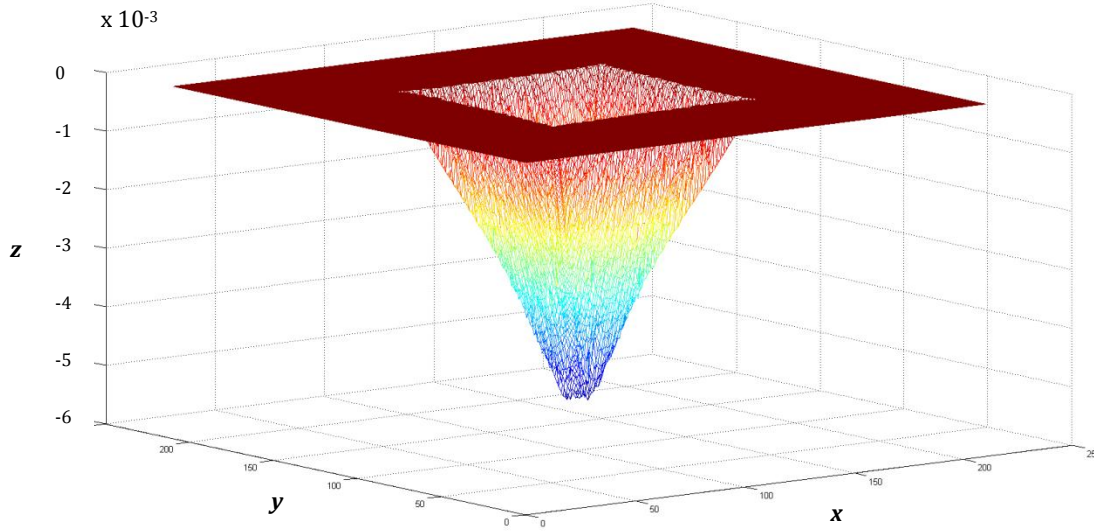


Figure 4.9 (a): Vertical infiltration profile in 3-D. The flow is over sand.

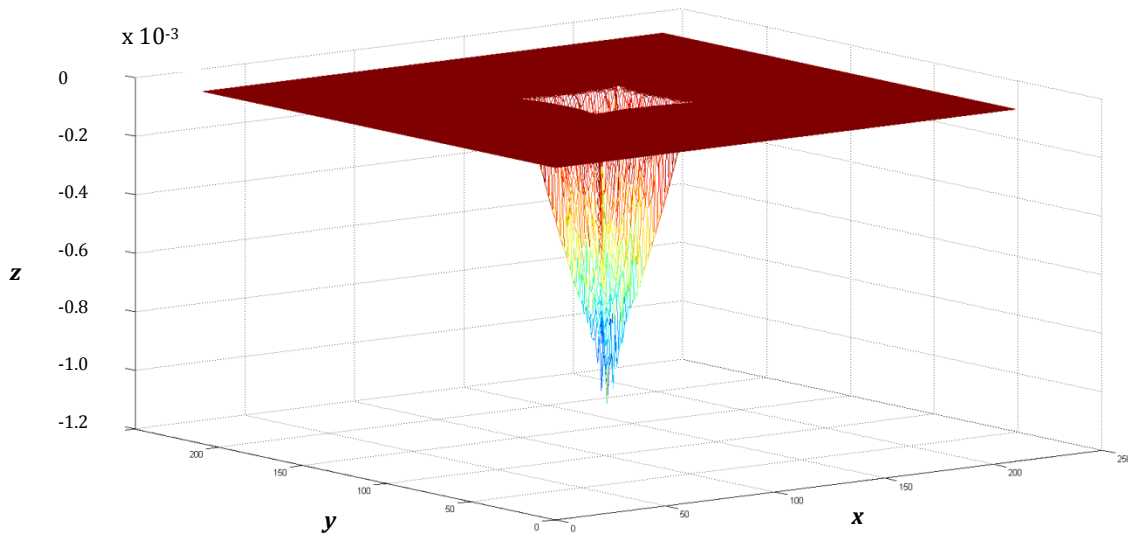


Figure 4.9 (b): Vertical infiltration profile in 3-D. The flow is over loam.

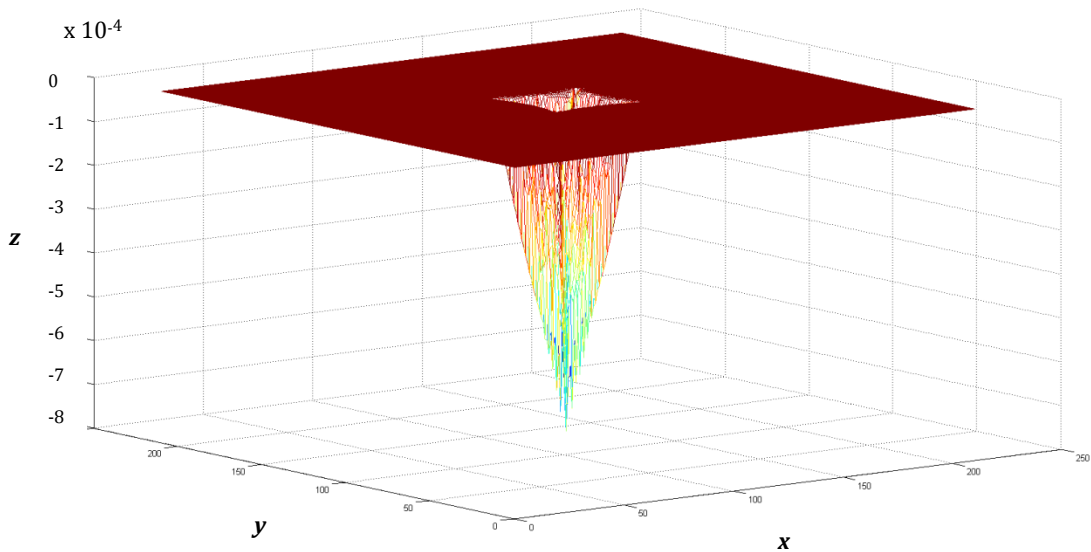


Figure 4.9 (c): Vertical infiltration profile in 3-D. The flow is over clay. Note the axis of this figure is to a different scale (10^{-4}) from those of Figs. 4.3 (a) and (b) which are 10^{-3}

Table 4.9: Comparing the relative values of the pore-size distribution index with the numerical results given for the rate of infiltration and cumulative infiltration depth for three different soil textures. The results show an expected trend of sand having a greater infiltration rate and cumulative infiltrative depth than loam which in turn is greater than clay.

	Sand	Loam	Clay
Pore-Size Distribution Index, λ	0.592	0.220	0.081
Rate of Infiltration, dl/dt (m/s)	13.7×10^{-5}	3.69×10^{-5}	2.85×10^{-5}
Cumulative Infiltration Depth, I (m)	0.0105	0.0029	0.0023

The results of the model again conform to what we know of vertical infiltration through soils. The rate of infiltration initially rises and as water supply continues and cumulative depth increases, it reduces and tends towards the hydraulic conductivity of the soil. Each

plot of the change of infiltration rate over time shows this trend. This shows that the infiltration model works well with the SWEs and that our numerical algorithm can handle problems of this nature.

4.10 Sediment Delivery

It is worth noting that in the case of a frictionless flat surface, the sediment load G in both directions is zero. This is because flow shear stress, τ_f , is due to bed friction, S_f , and hence it is zero when there is no friction.

Mathematically:

$$\tau_f = \gamma h S_f = 0 \quad (4.2)$$

since $S_f = 0$

$$T_c = k_t \tau_f^{\frac{3}{2}} = 0 \quad (4.3)$$

since $\tau_f = 0$

T_c is the sediment's transport capacity. From the flowchart in Figure 3.13, $G = 0$ is not greater than $T_c = 0$ hence no net deposition can take place. $\tau_f = 0$ is also not greater than the critical shear stress of the soil, τ_c , since $\tau_c > 0$ thus net detachment is set to zero and the update on G results in no change in value. Once again the model give result which accords to both physical and mathematical reasoning. We now turn to cases where there is bottom slope and friction slope in the computational domain.

4.11 Flow in a Real Terrain with Steep Gradients

Elevation data of a typical gas drilling location in Central Arkansas was obtained so that the performance of the model can be assessed. The elevation data is stored in a Digital Elevation Model (DEM). A DEM is a digital file consisting of terrain elevations for ground positions at regularly spaced horizontal intervals (USGS, 2012). The DEM used in this model is a 5 x 5 DEM which means each entry in the file represent the average elevation of a square area of ground whose side length is 5m. An area of $50,000m^2$ (250m by 200m) is randomly selected for analysis (Fig. 4.11a).

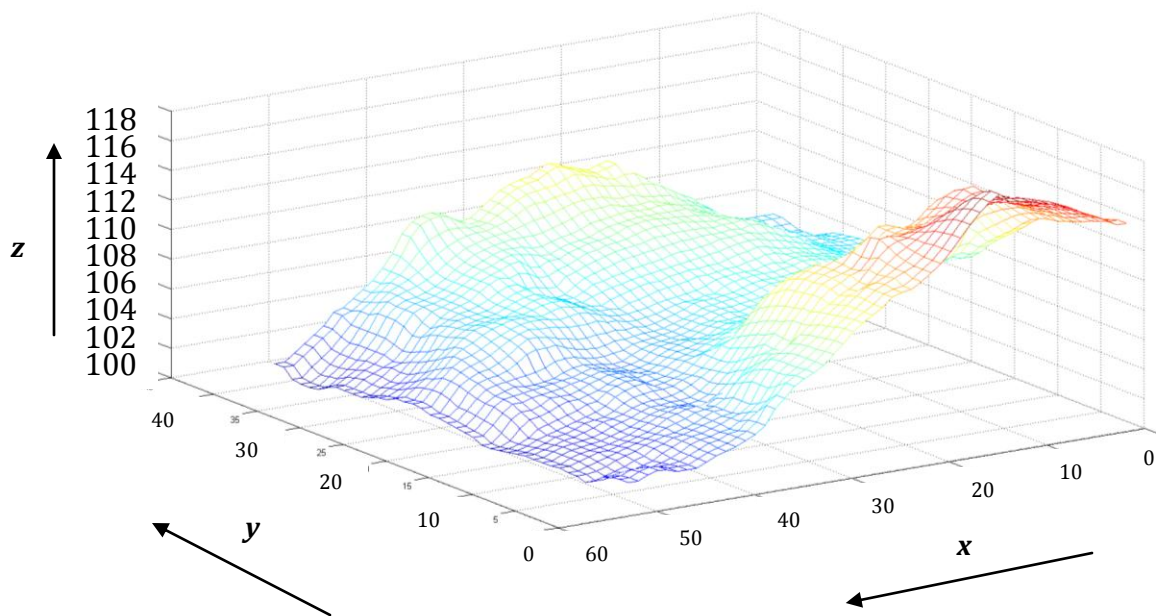


Figure 4.11 (a): A mesh plot of the elevation data obtained from the DEM of a gas drilling area in Central Arkansas. Terrains such as this have steep slopes that make solving the SWEs over them a vexing problem

Given our DEM size, this area corresponds to a computational array of 50 rows and 40 columns. The key features of this terrain are given below:

Highest point: 117.134m

Lowest Point: 100.032m

Steepest Slope in the x -direction, $S_{o,x}$: 0.3200

Steepest Slope in the y -direction, $S_{o,y}$: 0.2400

Steep gradients increase the non-linearity of coupled hyperbolic PDEs describing overland flow and make them less robust to solve numerically. This is why many works assume the kinematic or diffusive wave model when solving the SWEs. Even among those who employ the full dynamic form of the St. Venant equation, a small slope of about 0.01 is usually used to verify their models. To the best of the author's knowledge, this is the first work that will attempt to tackle a bed slope of this magnitude.

For the simulation carried out, the size of the time step, Δt , was 10^{-4} s. Such a small time step makes the computation run for a longer period but it also makes the solution grow in a bounded fashion and hence stable. The initial and boundary conditions are the same as those of the frictionless flat surface case. Some algorithms exist in the literature which claims to be able to solve the SWEs over a "dry" bed. On closer examination however, they always specify a minimum positive flow depth below which the bed is assumed dry. A starting value of $h = 0$ will simply lead to mathematical singularities.

The free surface profile of the flow after 4s is shown in Fig. 4.11(b). The flow is very gravity driven as per the source terms in the momentum equations:

$gh(S_o - S_f)$ and since we started with a high value of h , the friction slope term is relatively small compared to bed slope term.

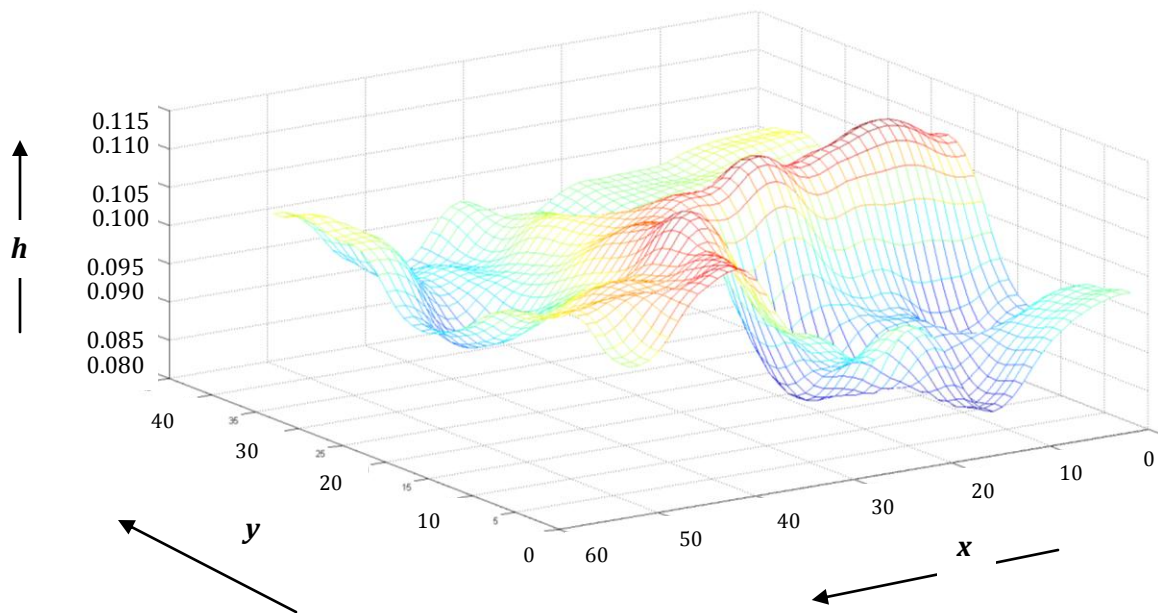


Figure 4.11(b): The free surface profile of the flow right after the start of the simulation ($t = 4s$). Water can be seen here quickly moving from high elevation areas (see Fig. 4.9) and flowing into the lowlands.

The free surface profile of the flow after 30s is shown next in Fig. 4.11(c). The figure shows that water has almost completely drained from the areas with higher elevation and is moving towards the boundaries through areas of lower elevation. One common misreading of free surface profiles is to see water as climbing uphill because an area of the 3-D plot is more elevated than another. A correct interpretation is that these show the depth of fluid on each cell. Cells that receive more fluid because of their elevation relative to their neighbors are expected to have greater fluid depth and thus appear higher in a 3-D profile.

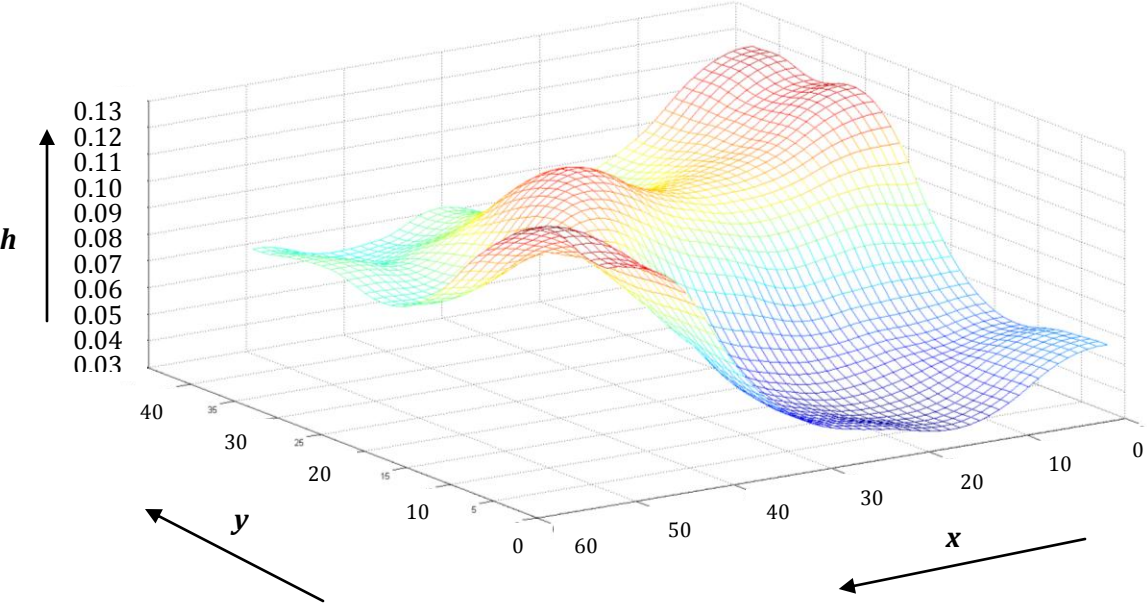


Figure 4.11(c): The free surface profile of the flow right after 30s. The high elevation areas are left with very little water while water continues to drain through areas of low elevation as it makes its way to the boundaries

Figs. 4.11 (d) and (e) show the result in 2-D. The DEM of the area and the flow depth are both shown in 2-D and placed side-by-side to aid in visualizing the results. The color gradient is such that red represents the highest values and blue represents the lowest values and there is a spectrum of colors to depict the various depths between these two extremes. The lower right corner of Fig. 4.11 (d) is where the highest elevation of the domain occurs. The corresponding area in Fig. 4.11 (e) has the lowest fluid depth because the water there has flowed downhill. The downstream boundary in both the x - and y -direction in Fig. 4.11 (d) has low elevation and fluid drains into and through it as seen in Fig. 4.11 (e). The lowest fluid depth of $0.0234m$ occurred at position (21, 8) in the computational domain. This is because the steepest descent in the entire domain is between cells (22, 8) and (21, 8).

The data values from these numerical experiments are too large to be put in a table and even if we do put them in a table, it will be hard for readers to make sense of them. This is why we have resorted to mesh plots of the data in explaining these results. Nevertheless we have included the data results in the Appendix for the sake of completeness and to help in further research.

The pronounced bed slopes in the terrain are such that flow was remarkably unidirectional. Water movement was always towards the $+x$ and $+y$ - direction. Fig. 4.11 (f) shows the velocity plot of the flow. All the values of velocities in both horizontal directions are positive.

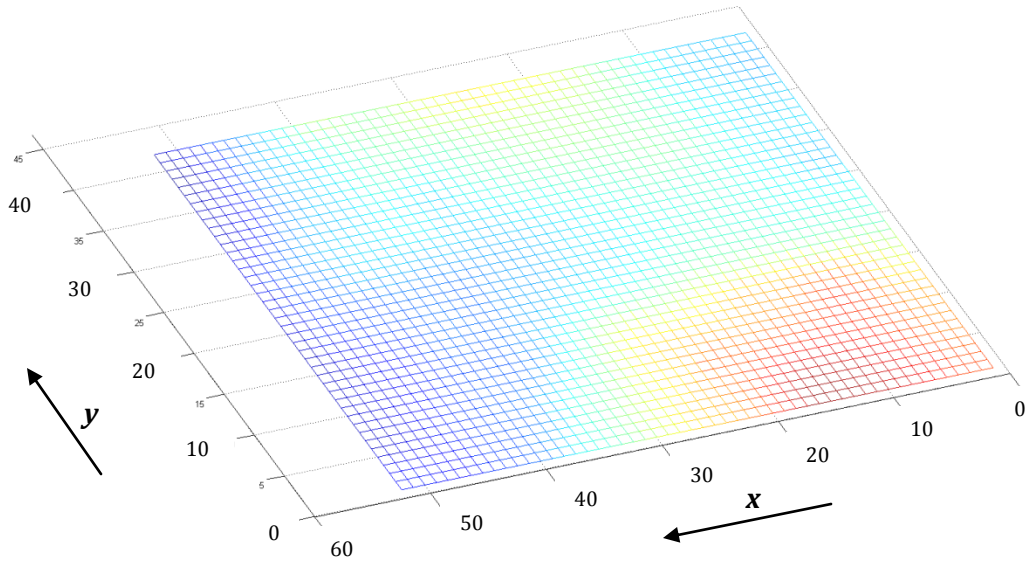


Figure 4:11(d): A 2-D view of the mesh plot of the DEM showing the relative elevation of the cells. Cells with high elevation are red and those with low elevations are blue.

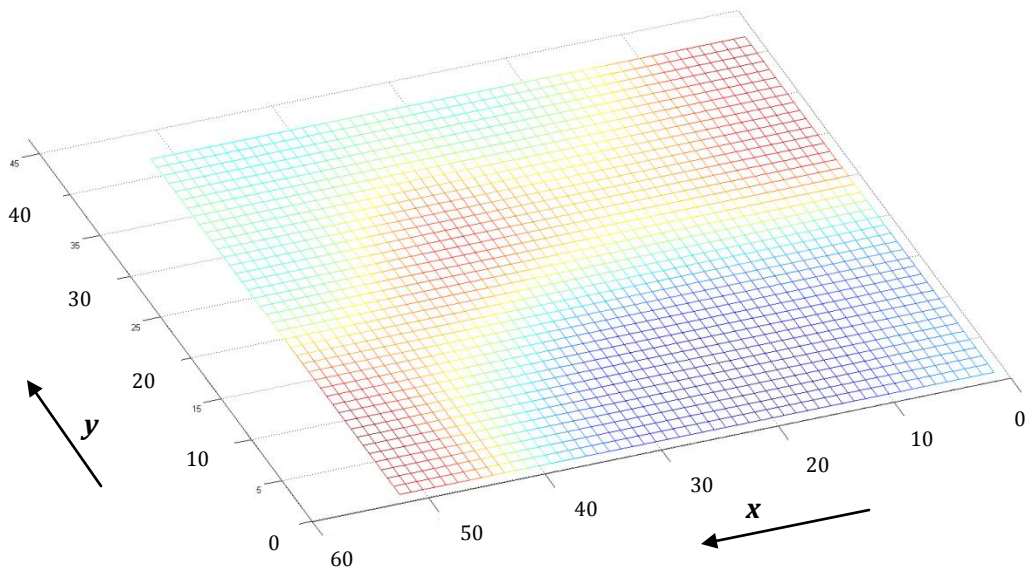


Figure 4:11(e): A 2-D view of the free-surface profile of the water depth after 30s of simulation. Notice how the low water depths correspond to the high elevation areas in Fig. 4.11 (d) above.

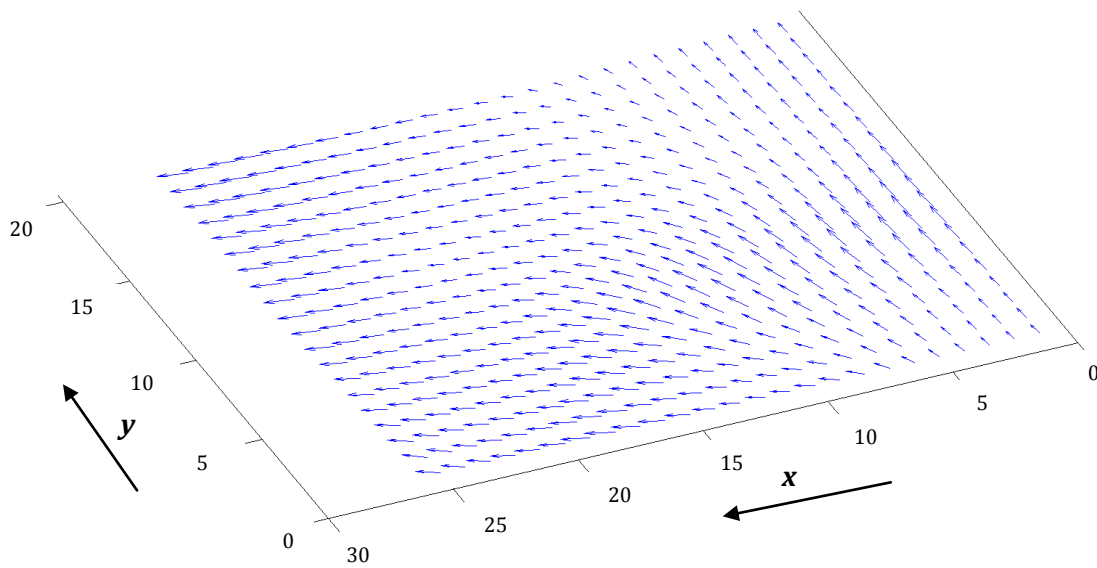


Figure 4.11(f): The flow field shows the magnitude and direction of the velocity. Increased momentum as flow moves downhill caused increase in the velocity as depicted by the longer arrows at and near the downstream boundary

Even though all the velocities in this example were positive, this should not be taken as a general rule. Velocities, unlike fluid depth or pollutant concentration, can take negative values. The negative sign indicates the direction of flow and has nothing to do with its magnitude. The size of the computational domain was reduced to a fourth of its original size using the procedure mentioned in Section 4.7. The reduction is necessary to enhance the visualization of the quiver plot. The quiver plot of the original domain was compared to the abridged version in Fig. 4.11(f) and the comparison showed that no significant information was lost using this procedure.

4.12 Pollutant Transport over Real Topography

We have shown that the model gives expected results despite the occurrence of steep gradients in the computational domain. The flow depth and velocity field are important because they help us calculate discharge and also because they are what the pollutant transport model and the erosion/sediment transport model relies on.

The pollutant transport model was tested for this problem using the following values:

Initial conditions:

$$C_0(x, y, 0) = 0$$

$$C_0(25, 20, 0) = 38 \text{ ppm}$$

After 10s:

$$C(25, 20, t > 10) = 0$$

Boundary conditions:

$$\frac{\partial C_x}{\partial x} = 0 \tag{4.4}$$

$$\frac{\partial C_y}{\partial y} = 0 \tag{4.5}$$

The zero-gradient concentration imposed at the boundary will allow pollutant species to leave the domain if they reach the downstream boundary. The result of this simulation is shown in Figs. 4.12. Fig. 4.12 (a) is a longitudinal profile of the pollutant concentration. The

position is at $y = 20$ or the center of the y -axis while Fig. 4.12 (b) is the 3-D plot of surface profile of pollutant concentration.

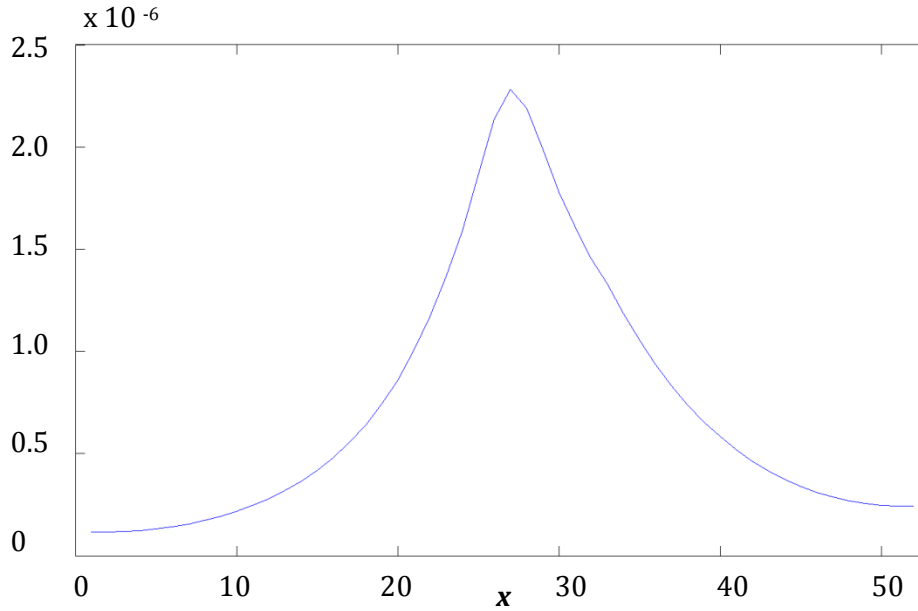


Figure 4.12 (a): A 1-D plot of pollutant concentration profile. The source of the pollutant is at the middle (the peak of the figure). Pollutant species is seen migrating towards the downstream as they are carried by the flow.

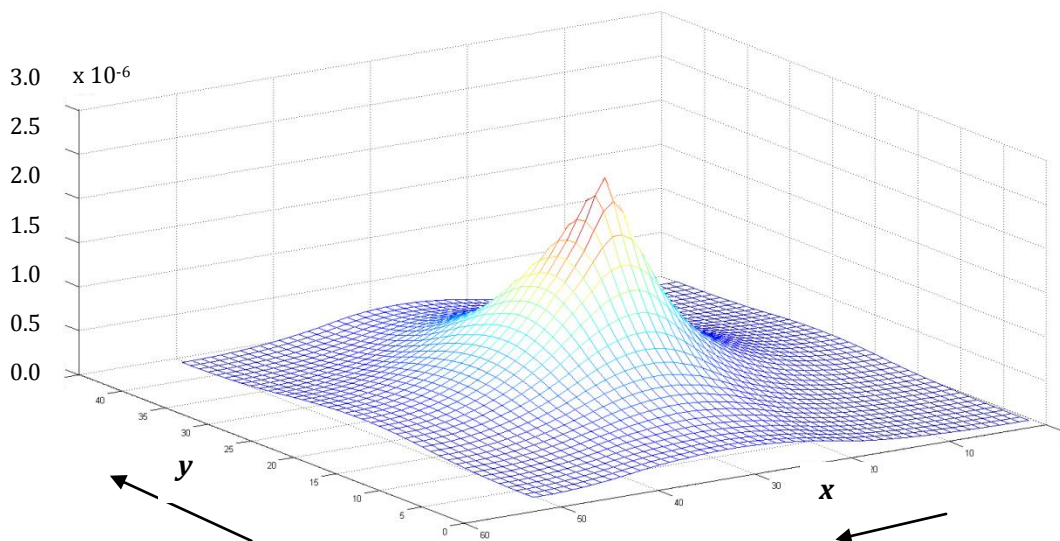


Figure 4.12 (b): A mesh plot of pollutant concentration profile. The profile is dictated by the flow field.

The plot in Fig. 4.12 (a) shows that the peak has reduced significantly from 38ppm to less than 2.5ppm in the time after the source was turned off. The pollutant flows towards the downstream boundary. As the velocity plot (Fig. 4.11f) indicated, flow was towards the +x and +y -direction. The concentration recorded at the upstream of the domain is due to round-off errors. In reality, there should be zero concentration at the upstream. The concentration profile was – as expected – dictated by the slopes in the terrain. Figure 4.12 (b) shows the 3-D plot of the concentration profile at the end of the 30s simulation. It was greatest at the source, virtually zero at the upslope and barely reaching the downslope during that period.

The flow in this simulation is mainly supercritical as most of the velocities have magnitudes greater than the critical velocity. There are some regions in the domain however that experienced subcritical flows. This is true for most practical flows: there will be regions of subcritical and supercritical flows and even a region with subcritical flow in one time instance may change to one with supercritical flow at another point in time. By giving reasonable results, the model has shown its ability to handle both subcritical and supercritical flows simultaneously.

4.13 Sediment Delivery over Real Topography

Erosion/sediment transport model was coupled with the SWEs in this simulation to give the sediment loads on each cell. Figs. 4.13 show the result of the simulation. Fig. 4.13 (a) shows the sediment delivery in the x-direction while fig. 4.13(b) shows the delivery in the

y -direction. The shape of the terrain (Fig. 4.11a) and the flow field (Fig. 4.11f) are key in interpreting the results of the erosion/sediment transport model.

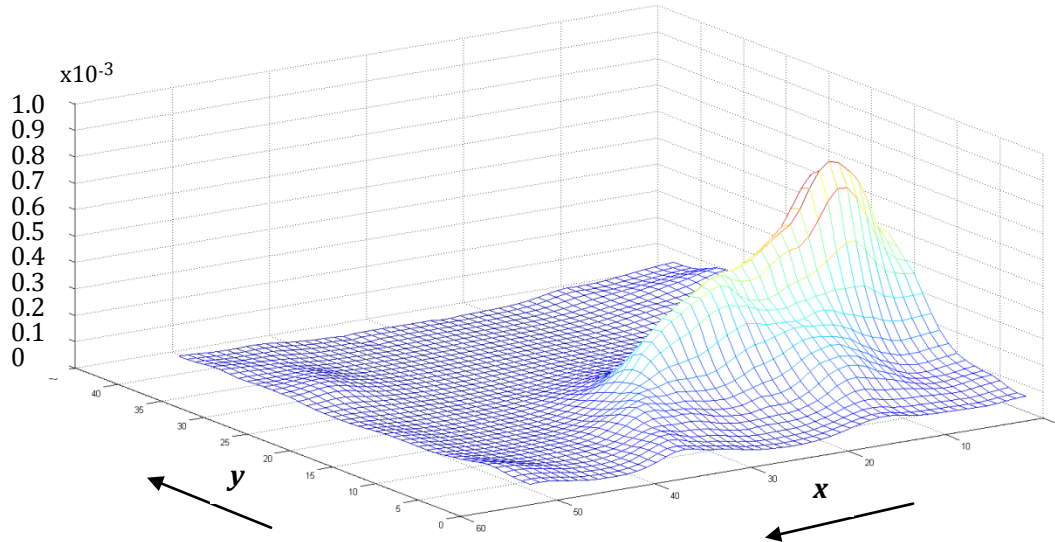


Figure 4:13(a): Mesh plot of Sediment load in the x -direction, G_x . The cells with the high loads are where the gradient is at the steepest hence more detachment of soil occurs there.

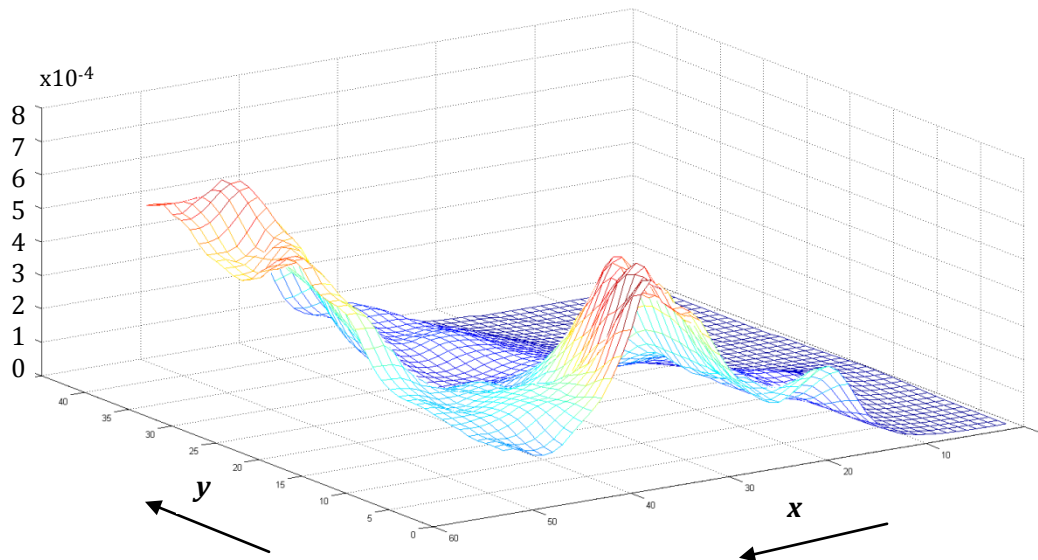


Figure 4:13(b): Mesh plot of Sediment load in the y -direction, G_y . The sediment delivery at the downstream boundary is greater because interior cells are pouring into it

The initial conditions were:

$$G_x(x,y,0) = 0 \text{ and}$$

$$G_y(x,y,0) = 0$$

Sediments can be detached, transported or deposited. These three processes can occur simultaneously within the same cell so that what we usually discuss – and what the model computes – are net detachment, net transport and net deposition. The steep gradients at the upstream cells in the x -direction cause rapidly varied flows which erode soil quickly. The sediment load from these cells is therefore relatively high. The highest value calculated was 0.0008 kg/m/s . The downstream in the x -direction also has significant sediment loads because of deposition from the interior cells. The total sediment discharge can be calculated by summing all the sediment loads from a boundary and multiplying by the width of the boundary and the length of time under consideration.

The model we have for solving SWEs has been tested both for a simple ideal surface and for a complex real terrain. We have examined its ability to handle gradually varied and rapidly varied flows. We also ran simulations to ensure that other models such as infiltration, pollutant transport and 2-D erosion/sediment transport was successfully coupled to the SWEs. In the next chapter, we will discuss the conditions under which the model can be expected to give bankable results as well as the limitations of the model. We will also give recommendations for further research in this area.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

This work addressed the problem of reserve pit failures encountered in gas drilling sites in the Fayetteville Shale Play. By knowing the potential impact of a spill following a pit failure, better management decisions can be made that will limit the potential environmental damage that can be caused. This work will help in the screening of suitable locations to locate a reserve pit and it can also help in shaping best management practices (BMPs) when it comes to oil and gas infrastructure development.

At the core of the decision support system we created is a mathematical model that describes the overland flow of fluid. An infiltration model that describes the vertical infiltration of water into the soil, a pollutant transport model that shows the spatial-temporal evolution of pollutant species concentration and an erosion/sediment transport model that gives the sediment delivery from a given event were coupled with the Shallow Water Equations (SWEs) that govern overland flow to give a more complete description of the flow dynamics we will encounter in reality.

Oil spills in marine environments have been well studied and there exist a number of spill models suitable for aquatic conditions. Terrestrial spill models – to the best of the author’s knowledge – do not exist. This work is a first in that regard. The SWEs together with the various models coupled with it form a system of highly non-linear hyperbolic PDEs that

only yield to numerical solutions. A number of novel approaches were developed to handle the non-linearities encountered in the model.

A novel semi-implicit approach was developed inside the MacCormack explicit scheme to deal with the “shallow depth problem.” At shallow depths, frictional forces tend to become very large giving unrealistically large values of velocities and negative fluid depths. The new approach developed in this work solved the problem so that when fluid depths are shallow, the frictional forces are correspondingly low. This closely matches the physics of the situation.

A new erosion/sediment transport model was derived from first principles and reasonable assumptions about the nature of the problem. The only 2-D surface runoff model in the literature is CASC2D developed by Julien & Saghafian (1991). Johnson & Julien (2000) added an upland erosion subroutine to CASC2D and called it CASC2D-SED. CASC2D-SED uses the kinematic wave approximation to the St. Venant’s equation in getting values of flow depth and flow velocities and also assumes gentle slopes (<5%) in the terrain. This project looked at terrains with steep slopes (>30%) and uses the fully dynamic form of the St. Venant’s equation to obtain the flow variables so, in that sense, it is better than CASC2D-SED.

The ability to handle very shallow water depths and highly steep slopes in a topography while giving information about flow discharge, areal distribution of pollutant concentration and sediment delivery simultaneously is the new knowledge this dissertation brings to the field of environmental protection and shale gas development. It is an advancement of the modeling capabilities on real topography for surface fluid transport of finite volume spills.

5.2 Limitations of the Model

1. Computational Speed

The MacCormack explicit scheme used in solving the set of equations describing the various aspects of the flow has been shown to be the best available scheme for such purposes (Hoffman, 1992). The scheme's major con though is that it is prone to instability. To assure a stable solution, a limit is placed on the size of time step that can be taken from one time level to the next. To successfully deal with the steepness encountered in the model, very small time steps ($\sim 10^{-4}s$) were taken. This gave a solution that is reliable and stable but it takes a long time for the program to return a result. The steeper the slopes, the smaller the time steps that must be taken.

2. Raindrop Impact

The impact of raindrops was neglected in formulating the erosion/sediment transport model. Raindrops loosen a compact soil making it more susceptible to detachment and so erosion models typically include a term to account for this phenomenon. The reason for neglecting this dynamic here is that this project is primarily focused on what happens in the event of a single incident involving a large spill. But if a reserve pit overflows and spills its contents because of a highly intense rainfall or if a spill incident coincides with a highly intense rainfall, then the assumption of zero raindrop impact becomes invalid. The model will thus underestimate sediment yield in these scenarios.

5.3 Recommendations

1. Computational Speed

To make the model useful for real-time decision making, its computational speed has to be greatly improved. There are two ways to accomplish this.

First, the computer program should be translated from MATLAB into a faster programming language. The computer model was written in MATLAB because it is an ideal language for developing and debugging programs. MATLAB prioritizes usability over optimal program efficiency. To obtain results on time however, a much faster and more efficient optimized code should be used.

Second, the program's backend calculations need to be run on a supercomputer with powerful processor speeds and bigger memory. The user will input values from a desktop but the real computations will be taking place on the supercomputer. Other fast computing alternatives can be explored, for example, parallel micro-computers may be used instead of one big supercomputer.

2. Erosion/Sediment Transport Model

The default value of the soil erodibility parameter, k_t , and the soil critical shear stress, τ_c , were taken from the Water Erosion Prediction Project (WEPP) model. WEPP arrived at the value of $0.0115s/m$ for k_t and $3.1Pa$ for τ_c by assuming a soil sample with equal ratio of different soil textures and by assuming moderate vegetative cover and soil organic content. If the soil texture in a particular application is known, and some other information about

the vegetation and soil organic content is also known, the k_t and τ_c values for that specific application should be used in place of the default. Appendix G gives the guideline on how to determine the k_t and τ_c value for a given set of soil conditions.

The model neglects the impact of raindrops as a detachment process. If the land is bare with no vegetative cover or if it is the site of a new construction with lots of loose soils and a storm event occurs, the actual sediment yield will be significantly greater than that predicted by the model. One way to ameliorate this problem is to increase the value of k_t and reduce the value of τ_c so that soil detachment can be greater. The increased detachment can then be seen as resulting from raindrop impact.

5.4 Areas of Further Research

1. Continued research is necessary in the area of computational speed. Ways to make the program run faster should be investigated. We developed a model that is as parsimonious as possible. We captured the key (but not all of the) physics that can occur in a spill event. As the model becomes more complex to closely match reality, the need for a faster program will become even more important.
2. The pollutant transport model should be improved so that it can handle chemical reaction among the pollutant species or between the species and soil sediments. It should also be able to treat pollutant (biological) decay and adsorption onto soil particles.
3. The erosion/sediment transport model can only address one class of soil texture at a time. The model should be modified so that it can handle a mixture of soil classes.

The ratio of each soil class would be known and the soil mixture would have an aggregate soil properties value that would be a mean of the different values of its constituent parts.

REFERENCES

- Abbott, M. B. (1966). *An introduction to the method of characteristics*. New York: Elsevier.
- Abderrezzak, K. Paquier, A. (2011). Applicability of sediment transport capacity formulas to dam-break flows over movable beds. *Journal of Hydraulic Engineering*, 137(2) 209.
- Akan, A. O., & Yen, B. C. (1981). Mathematical model of shallow water flow over porous media. *J. Hydraul. Div. Am. Soc. Civ. Eng.*, 107(4), 479-494.
- Alonso, C. V., Neibling, W. H., & Foster, G. R. (1981). Estimating sediment transport capacity in watershed modeling. *Trans. ASAE*, 24(5), 1211-1220, 1226.
- American Society of Civil Engineers. (1963). Friction factors in open channels - progress report of the task committee on hydrodynamics of the hydraulics division. *J. Hydraul. Div. Am. Soc. Civ. Eng.*, 89, 97-143.
- Anderson, D.A., Tannehill, J.D. & Pletcher, R.H. (1984). *Computational fluid mechanics and heat transfer*. New York, NY: McGraw-Hill.
- Annual energy outlook 2011(2011). U.S. Energy Information Administration.
- Arcement, G. J. J., & Schneider, V. R. *Guide for selecting manning's roughness coefficients for natural channels and flood plains*. Retrieved February, 13, 2012, from <http://www.fhwa.dot.gov/bridge/wsp2339.pdf>
- Assouline, S. (2005). On the relationships between the pore size distribution index and characteristics of the soil hydraulic functions. *Water Resour. Res.*, 41, W07019, doi: 10.1029/2004WR003511.
- Barry, D. A., Parlange, J. Y., Haverkamp, R., & Ross, P. J. (1995). Infiltration under ponded conditions: 4. an explicit predictive infiltration formula. *Soil Science*, 160, 8-17.
- Basco, D. R., (1983). *Computation of rapidly varied, unsteady free-surface flow* No. WRI 83-4284.
- Beasley, D. B., Huggins, L. F., & Monke, E. J. (1980). ANSWERS--a model for watershed planning. *Trans. ASAE*, 23, 938-944.
- Beck, M. B. (1987). Water quality modeling: A review of uncertainty. *Water Resources Research*, 23(8), 1393-1442.
- Bennett, J. P. (1974). Concepts of mathematical modelling of sediment yield. *Water Resources Research*, 10, 485-492.

- Bhar, R., & Nikolova, B. (2009). Oil prices and equity returns in the BRIC countries. *The World Economy*, 32(7), 1036-1054.
- Brakensiek, D. L. (1977). Estimating the effective capillary pressure in the green and ampt infiltration equation. *Water Resources Research*, 13(3), 680-682.
- Bull, L. J., & Kirkby, M. J. (1997). Gully processes and modelling. *Progress in Physical Geography*, 21(3), 354-374.
- Burden, R. L., & Faires, J. D. (2011). *Numerical analysis* (9th ed.) Brooks/Cole, Cengage Learning.
- Chaudhry, M.H. (1993). *Open-channel flow*. Englewood Cliffs, NJ: Prentice-Hall, 279-282.
- Chow, V.T. (1959). *Open channel hydraulics*, New York, NY: McGraw-Hill.
- Christensen, B. A. (1985). Open channel and sheet flow over flexible roughness. *21st IAHR Congress, Int. Assoc. Hydraul. Res.* Melbourne, Victoria, Australia.
- Ciesolka, C. A., Coughlan, K. J., Rose, C. W., Escalante, M. C., Hashim, G. M., Paningbatan, E. P. J., et al. (1995). Methodology for a multi-country study of soil erosion management. *Soil Technology*, 8, 179-192.
- Clark, L. A. & Wynn, T. M. (2007). Methods for determining stream-bank critical shear stress and soil erodibility: implications for erosion rate predictions. *Transactions of the ASABE*, 50 (1), 95- 106.
- Clausnitzer, V., Hopmans, J. W., & Starr, J. L. (1998). Parameter uncertainty analysis of common infiltration models. *Soil Sci. Soc. Am. J.*, 62, 1477-1487.
- Coleman, J. (2011). Tight-gas sandstone reservoirs: The 200-year path from unconventional to conventional gas resource and beyond. *Unconventional Energy Resources: Making the Unconventional Conventional*, 1, 397-441.
- Compton, R. R. (1985). *Geology in the field*. New York, NY: John Wiley and Sons.
- Cooley, R. L., & Moin, S. A. (1976). Finite element solution of Saint-Venant equations. *J. Hydraul. Div. Am. Soc. Civ. Eng.*, 102(HY6), 759-775.
- Courant, R., Friedrichs, K., & Lewy, H. (1928). *On the partial difference equations of mathematical physics* No. NYO-7689)AEC Research and Development Report.
- Cowan, W. L. (1956). Estimating hydraulic roughness coefficients. *Agricultural Engineering*, 37(7), 473-475.

- Cox, L. A. J., & Ricci, P. F. (2006). Reassessing benzene cancer risks using internal doses. *Risk Analysis*, 12(3), 401.
- CSIRO TOPOG. Retrieved February/17, 2012, from <http://www.clw.csiro.au/topog/intro/intro.html>
- Cunge, J. A., Holly, J. F. M., & Verwey, A. (1980). *Practical aspects of computational river hydraulics*. London: Pitman Publications.
- Cupas, A. C. (2009). The not-so-safe drinking water act: Why we must regulate hydraulic fracturing at the federal level. *Williamand Mary Environmental Law and Policy Review*, 33(2), 605.
- Daubert, A. & Graffe, O. (1967). Quelques aspects des écoulements presque horizontaux a deux dimensions en plan et non permanents, *La Houille blanche*, 8, 847-860.
- Davis, S. S. (1978). Deposition of nonuniform sediment by overland flow on concave slopes. (Masters, Purdue University, W. Lafayette, IN).
- de Saint-Venant, B. (1871). Theorie du mouvement non-permanent des eaux. *Acad. Sci. Compts Rendus*, 73, 148-154, 237-246.
- Department of Land and Water Conservation. (1995). *IQQM-integrated water quality and quantity model* No. TS95.019) Catchment Processes and Modelling Branch.
- Dooge, J. C. I. (1989). The manning formula in context. *Proceedings of the International Conference for Centennial of Manning's Formula and Kuichling's Rational Method*, University of Virginia, Charlottesville. pp. 849-903.
- Dronkers, J.J. & Schonfeld, J.C. (1955). Tidal computations in shallow water. *Am. Soc. Civil Engineers Proc.* 81(714), 49.
- Ebel, B. A., Hinckley, E. S., & Martin, D. A. (2012). Soil-water dynamics and unsaturated storage during snowmelt following wildfire. *Hydrology and Earth System Sciences Discussions*, 9, 441-483.
- Egan, N. W. (2012). Where the water catches fire. *People*, 77(5), 72.
- Einstein, H. A. (1968). Deposition of suspended particles in a gravel bed. *J. Hydraul. Div. Am. Soc. Civ. Eng.*, 94(HY5), 1197-1205.
- Elder, J. W. (1959). The dispersion of marked fluid in turbulent shear flow. *Journal of Fluid Mechanics*, 5(4), 544-560.

- Esteves, M., Faucher, X., Galle, S., & Vauclin, M. (2000). Overland flow and infiltration modeling for small plots during unsteady rain: Numerical results versus observed values. *Journal of Hydrology*, 228, 265-282.
- Falconer, R. A. (1991). Review of modelling flow and pollutant transport processes in hydraulic basins. In L. C. Wrobel, & C. A. Brebbia (Eds.), *Water pollution: Modelling, measuring and prediction* (pp. 1-23) Computational Mechanics Publications & Elsevier Applied Science.
- Fennema, R. J., & Chaudhry, M. H. (1990). Explicit methods for two-dimensional unsteady free-surface flows. *J. Hydraul. Div. Am. Soc. Civ. Eng.*, 116(8), 1013-1034.
- Fingas, M. (2010). Introduction to spill modeling. *Oil spill science and technology* (pp. 187-200). Burlington, MA: Gulf Professional Publishing, Elsevier.
- Fink, J. K. (2003). *Oilfield chemicals*. New York: Gulf Professional Publishing.
- Finkner, S. C., Nearing, M. A., Foster, G. R., & Gilley, J. E. (1989). A simplified equation for modeling sediment transport capacity. *Trans. ASAE*, 32(5), 1545-1550.
- Foster, G. R. (1982). Modeling the erosion process. In C. T. Haan, H. P. Johnson & D. L. Brakensiek (Eds.), *Hydrologic modelling of small watersheds* (pp. 297-370). St. Joseph, Michigan: American Society of Agricultural Engineers.
- Foster, G. R., Flanagan, D. C., Nearing, M. A., Lane, L. J., Risse, L. M., & Finkner, S. C. (1995). Chapter 11: Hillslope erosion component. *WEPP manual* ()
- Foster, G. R., Lane, L. J., Nowlin, J. D., Laflen, J. M., & Young, R. A. (1981). Estimating erosion and sediment yield on field-sized areas. *Trans. ASAE*, , 1253-1262.
- Foster, G. R., & Meyer, L. D. (1972). A closed-form soil erosion equation for upland areas. In H. W. Shen (Ed.), *Sedimentation symposium to honor Prof. H.A. Einstein* (pp. 12.1-12.9). Fort Collins, Colorado.
- Friedman, T. (2008). Clean electrons. *Hot, flat and crowded. why we need a green revolution –and how it can renew America* (1st ed., pp. 186-186). New York: Farrar, Straus and Giroux.
- Garcia, R., & Kahawita, R. A. (1986). Numerical solution of the st. venant equations with the MacCormack finite-difference scheme. *International Journal for Numerical Method in Fluids*, 6(5), 259-274.
- Ghosh, R. K. (1980). Modeling infiltration. *Soil Science*, 130, 297-302.

- Govindaraju, R. S., Jones, S. E., & Kavvas, M. L. (1988a). 1. on the diffusion wave model for overland flow 1. solution for steep slopes. *Water Resources Research*, 24(5), 734-744.
- Govindaraju, R. S., & Kavvas, M. L. (1991). Modeling the erosion process over steep slopes: Approximate analytical solutions. *Journal of Hydrology, Amsterdam*, 127, 279-305.
- Green, W. H., & Ampt, G. A. (1911). Studies on soil physics: 1. the flow of air and water through soils. *J. Agric. Sci. (Cambridge)*, 4, 1-24.
- Groundwater Protection Council. (April 2009). *Modern shale gas development in the united states: A primer*. Oklahoma City, OK: U.S. Department of Energy, National Energy Technology Laboratory with ALL Consulting.
- Groves, J. R. (1989). A practical soil moisture profile model. *Journal of the American Water Resources Association*, 25(4), 875-880.
- Guermond, J. L., Pasquetti, R. & Popov, B. (2011). Entropy viscosity method for nonlinear conservation laws, *J. Comput. Phys.*, 230, 4248-4267.
- Gupta, D. V. S., & Hlidek, B. T. (2009). Frac fluid recycling and water conservation: A case history. *SPE Hydraulic Fracturing Technology Conference*, The Woodlands, TX.
- Gutteridge Haskins and Davey. (1991). *Integrated Quantity/Quality modelling--stage 3*. Department of Water Resources, Sydney: Gutteridge Haskins and Davey.
- Guymer, I., Wilson, C. A. M. E., & Boxall, J. B. (2005). Modelling solute transport processes in free surface flow CFD schemes. In P. D. Bates, S. N. Lane & R. I. Ferguson (Eds.), *Computational fluid dynamics: Applications in environmental hydraulics* () John Wiley & Sons, Ltd.
- Haan, C. T., Barfield, B. J., & Hayes, J. C. (1994). *Design hydrology and sedimentology for small catchments* Academic Press.
- Hadamard, J. (1923). *Lectures on cauchy's problem in linear partial differential equations* Dover Publications.
- Hairsine, P. B., & Rose, C. W. (1991). Rainfall detachment and deposition: Sediment transport in the absence of flow-driven processes. *Soil Sci. Soc. Am. J.*, 55(2), 320-324.
- Hairsine, P. B., & Rose, C. W. (1992a). Modelling water erosion due to overland flow using physical principles: 2. rill flow. *Water Resources Research*, 28(1), 245-250.
- Hanley, N., Faichney, R., Munro, A., & Shortle, J. S. (1998). Economic and environmental modelling for pollution control in an estuary. *Journal of Environmental Management*, 52, 211-225.

- Hanson, G. J. (1990a). Surface erodibility of earthen channels at high stresses: Part I. Open channel testing. *Trans. ASAE* 33(1):127-131.
- Hanson, G. J. (1990b). Surface erodibility of earthen channels at high stresses: Part II. Developing an in situ testing device. *Trans. ASAE* 33(1): 132-137.
- Hartmann, R. & Houston, P. (2002). Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183, 508-532.
- Hayter, E. J., & Pakala, C. V. (1989). Transport of inorganic contaminants in estuarial waters. *Journal of Coastal Research*, (5), 217-230.
- Henderson, F. M. (1966). *Open-channel flows*. New York: MacMillan.
- Henderson, F. M., & Wooding, R. A. (1964). Overland flow and groundwater flow from a steady rainfall of finite duration. *J. Geophys. Res.*, 69, 1513-1540.
- Hicks, F. E., & Steffler, P. M. Comparison of finite element methods for the st. venant equations. *International Journal for Numerical Method in Fluids*, 20(2), 99-113.
- Hillel, D. (1971). *Soil and water: Physical principles and processes* Academic Press.
- Hjelmelt, A. T. (1981). Overland flow from time distributed rainfall. *Journal of Hydraulic Engineering*, 107(HY2), 227-238.
- Hoffman, J. D. (1992). *Numerical methods for engineers and scientists* McGraw Hill.
- Horton, R. E. (1938). The interpretation and application of runoff plot experiments with reference to soil erosion problems. *Soil Sci. Soc. Am. Proc.*, 3, 340-349.
- Horton, R. E. (1940). An approach towards a physical interpretation of infiltration capacity. *Soil Sci. Soc. Am. Proc.*, 5, 399-417.
- Howarth, R. W., Ingraffea, A., & Engelder, T. (2011). Natural gas: Should fracking stop? *Nature*, 477(7364), 271-275.
- Huang, S. L. (2009). Two-dimensional numerical modeling of chemical transport-transformation in fluvial rivers: Formulation of equations and physical interpretation. *Journal of Hydroinformatics*, 11(2)
- Hudson, N. W. (1975). The factors determining the extent of soil erosion. In R. Gremland (Ed.), *Soil conservation and management in the humid tropics* () John Wiley and Sons.
- Iwasa, Y., & Inoue, K. (1988). Mathematical simulations of channel and overland flood flows in view of flood disaster engineering. *Journal of Natural Disaster Science*, 4(1), 1-30.

- Jakeman, A., Post, D., & Beck, M. (1994a). From data and theory to environmental model: The case of rainfall runoff. *Environmetrics*, 5, 297-300.
- Jakeman, A.J. and Hornberger, G.M. (1993). How much complexity is warranted in a rainfall-runoff model? *Water Resources Research* 29(8): doi: 10.1029/93WR00877. ISSN: 0043-1397.
- Jakeman, A., Post, D., Schreider, S., & Yu, Y. W. (1994b). Modelling environmental systems: Partitioning the water balance at different catchment scales. In P. Zannetti (Ed.), *Computer techniques in environmental studies V. computational mechanics publications* (pp. 157-170) Southampton.
- Jameson, A., Schmidt, W., and Turkel, E. (1981). Numerical Solutions of the Euler equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. *AIAA 14th Fluid and Plasma Dynamics Conference*, Palo Alto, CA, AIAA-81-1259.
- Johanson, R. C., Imhoff, J. C., & Davis, H. H. (1980). *Users manual for the hydrologic simulation program--fortran (HSPF) version no. 5.0*. Athens, GA: USA EPA Environmental Research Laboratory.
- Johnson, B. E., & Julien, P. Y. (2000). The two-dimensional upland erosion model CASC2D-SED. *The Hydrology-Geomorphology Interface: Rainfall, Floods, Sedimentation, Land use*, Jerusalem. pp. 107-125.
- Johnson, H., & Dore, A. G. (2010). Unconventional oil and gas resources and the geological storage of carbon dioxide: Overview. *Petroleum Geology: From Mature Basins to New Frontiers*, 7. pp. 1061-1063.
- Julien, P.Y. & Saghafian, B. (1991). A two-dimensional upland erosion model. *Center for Geosciences—Hydrologic Modelling Group, Colorado State University (CER90-91PYJ-BS-12)*. Fort Collins, Colorado, USA.
- Kadlec, R. H. (1990). Overland flow in wetlands: Vegetation resistance. *Journal of Hydraulic Engineering*, 116, 691-706.
- Kao, D. T. Y., & Barfield, B. J. (1978). Prediction of flow hydraulics for vegetated channels. *Trans. ASAE*, 21, 489-494.
- Kawahara, M., & Yokoyama, T. (1980). Finite element method for direct runoff flow. *Journal of Hydraulic Engineering*, 106(HY4), 519-534.
- Knisel, W. G. (1980). *CREAMS: A field scale model for chemicals, runoff and erosion from agricultural management systems* USDA.
- Kurganov A. & Lin, C. T. (2007). On the reduction of numerical dissipation in central-upwind schemes. *Commun. Comput. Phys.*, 2, 141-163.

- Laflen, J. M., Lane, L. J., & Foster, G. R. (1991). WEPP: A new generation of erosion prediction technology. *Journal of Soil and Water Conservation*, 46, 34-38.
- Lai, C. (1965a). Flows of homogeneous density in tidal reaches, solution by method of characteristics. *U.S. Geological Survey Open-File Rep*,
- Lai, C. (1986). *Numerical modeling of unsteady open-channel flow. Advances in Hydrosience*, 14, 162-323.
- Laney, Bert. (2001). www.cfd-online.com/Forums/main/3676-artificial-viscosity.html. Retrieved on April 15, 2012.
- Lax, P.D. & Wendroff, B. (1960). Systems of Conservation Laws, *Comm Pure and Appl. Math*, 2, 159-193.
- Lei, T.W., Zhang, Q.W., Yan, L.J., Zhao, J., & Pan, Y.H. (2008). A rational method for estimating erodibility and critical shear stress of an eroding rill. *Geoderma*, 144 (3-4), 628-633.
- Le Roux, D. Y., Staniforth, A., & Lin, C. A. (1998). Finite elements for shallow-water equation ocean models. *Monthly Weather Review*, 126, 1931-1951.
- Leuterman, A. J. J., Jones, F. V., & Candler, J. E. (1988). Drilling fluids and reserve pit toxicity *Journal of Petroleum Technology*, 40(11), 1441.
- Liang, D., Binliang, L., & Falconer, R. A. (2007). Simulation of rapidly varying flow using an efficient TVD-MacCormack scheme. *International Journal for Numerical Method in Fluids*, 53, 811-826.
- Liang, D., Wang, X. & Falconer, R.S and Bockelmann Evans, B.N. (2010). *Solving the depth-integrated solute transport equation with a TVD-MacCormack scheme*. Environmental Modelling and Software, 25. Pp. 1619-1629, ISSN 1364-8152.
- Liggett, J. A., & Woolhiser, D. A. (1967) Difference solutions of shallow-water equation. *J. Eng. Mech. Div. Am. Soc. Civ. Eng.*, 93(EM2), 39-71.
- Littleboy, M., Silburn, M. D., Freebairn, D. M., Woodruff, D. R., Hammer, G. L., & Leslie, J. K. (1992b). Impact of soil erosion on production in cropping systems I. development and validation of a simulation model. *Australian Journal of Soil Research*, 30, 757-774.
- Loch, R. J., & Silburn, D. M. (1996). Constraints to sustainability--soil erosion. In L. Clarke, & P. B. Wylie (Eds.), *Sustainable crop production in the sub-tropics: An Australian perspective* () QDPI.
- MacCormack, R.W. (1969). The effect of viscosity in hypervelocity impact cratering. *Amer. Inst. Aero. Astro.*, Paper 69-354, Cincinnati, OH.

- MacDonald, I., Baines, M.J., & Nichols, N.K. (1995) Steady Open Channel Test Problems with Analytic Solutions. Numerical Analysis Report 3/95, Department of Mathematics, University of Reading, UK
- Maheshwari, B. L. (1992). Suitability of different flow equations and hydraulic resistance parameters for flows in surface irrigation: A review. *Water Resources Research*, 28(8), 2059-2066.
- Managing industrial solid wastes from manufacturing, mining, oil and gas production, and utility coal combustion* (1992). (Background Paper No. OTA-BP-O-82). Washington, D.C.: US Congress, Office of Technology Assessment.
- Manning, R. (1889). On the flow of water in open channels and pipes. *Trans. Inst. Civ. Eng. (Ireland)*, 20, 161-207.
- Massau, J. (1889). L'intégration graphique, and appendice au memoire sur l'integration graphique. *Assoc. Des Ingenieurs Sortis Des Ecoles Speciales De Grand (Belgium) Annales*, 12, 185-444.
- Massau, J. (1900). Graphical integration of partial differential equations. [Memoire sur l'integration graphique des equations aux derivees partielles] *Assoc. Des Ingenieurs Sortis Des Ecoles Speciales De Grand (Belgium) Annales*, 23, 95-214.
- Merritt, W. S., Letcher, R. A., & Jakeman, A. J. (2003). A review of erosion and sediment transport models. *Environmental Modelling & Software*, 18, 761-799.
- Meselhe, E. A., & Holly, J. F. M. (1993). Simulation of unsteady flow in irrigation canals with a dry bed. *Journal of Hydraulic Engineering*, 119, 1021-1039.
- Mingham, C. G., & Causon, D. M. (2008). A simple high-resolution advection scheme. *International Journal for Numerical Method in Fluids*, 56, 469-484.
- Misra, R. K., & Rose, C. W. (1996). Application and sensitivity analysis of process-based erosion model GUEST. *European Journal of Soil Science*, 47, 593-604.
- Morel-Seytoux, H. J., & Khanji, J. (1974). Derivation of an equation of infiltration. *Water Resources Research*, 10, 795-800.
- Morris, E. M., & Woolhhiser, D. A. (1980). Unsteady one-dimensional flow over a plane: Partial equilibrium and recession hydrographs. *Water Resources Research*, 16(2), 355-360.
- Murillo, J., Garcia-Navarro, P., Burguete, J., & Brufau, P. (2006). A conservative 2D model of inundation flow with solute transport over dry bed. *International Journal for Numerical Method in Fluids*, 52, 1059-1092.

- Nearing, M. A., Foster, G. R., Lane, L. J., & Finkner, S. C. (1989). A process-based soil erosion model for USDA water erosion prediction project technology. *Trans. ASAE* 32(5): 1587-1593.
- Nearing, M. A., Lane, L. J., & Lopes, V. L. (1994). Modelling soil erosion. In R. Lad (Ed.), *Soil erosion: Research methods*, 127-156.
- Osborn, S. (2012). Frack job. *Alternatives Journal*, 38(1), 5.
- Osman, A. M., & Thorne, C. R. (1988). Riverbank stability analysis: I. Theory. *J. Hydraulic Eng.* 114(2), 134-150.
- Parlange, J. Y., Rose, C. W., & Sander, G. (1981). Kinematic flow approximation of runoff on a plane: An exact analytical solution. *Journal of Hydrology*, 52, 43-58.
- Phillip, J. R. (1973). On solving the unsaturated flow equation: 1. the flux-concentration relation. *Soil Science*, 116, 328-335.
- Ponce, V. M., Ruh-Ming, L., & Simons, D. B. (1978). Applicability of kinetic and diffusion models. *Journal of Hydraulic Engineering*, 104, 353-360.
- Powers, B. (2011). *The Fayetteville shale peaks*. Retrieved February, 06, 2012, from <http://www.financialsense.com/contributors/bill-powers/2011/05/02/the-fayetteville-shale-peaks>
- Profile of the oil and gas extraction industry* (2000). No. EPA/310-R-99-006). Washington, D.C.: U.S. Environmental Protection Agency, Office of Compliance.
- Prosser, I. P., & Rustomji, P. (2000). Sediment transport capacity relations for overland flow. *Progress in Physical Geography*, 24(2), 179-193.
- Prosser, I. P., Young, B., Rustomji, P., Hughes, A., & Moran, C. (2001c). A model of river sediment budgets as an element of river health assessment. *Proceedings of the International Congress on Modelling and Simulation (MODSIM'2001)*, pp. 861-866.
- Ralston, A., & Rabinowitz, P. (1978). *A first-course in numerical analysis* (2nd ed.). New York: McGraw-Hill.
- Rao, S. S. (1999). *The finite element method in engineering* (3rd ed.). Boston: Butterworth-Heinemann.
- Rao, S. S. (2002). *Applied numerical methods for engineers and scientists* (1st ed.). Upper Saddle River, NJ: Prentice-Hall.
- Rawls, W. J., Brakensiek, D. L., & Saxton, K. B. (1981). *Soil water characteristics*. Chicago, Illinois: American Society of Agricultural Engineers.

- Recent advances in hydraulic fracturing*(1990). In Gidley J. L., Holditch S. A., Nierode D. E. and Veatch R. W. (Eds.), SPE.
- Reddy, J. N. (1993). *An introduction to the finite element method* (2nd ed.). New York: McGraw-Hill.
- Ree, W. O., Wimberley, F. L., & Crow, F. R. (1977). Manning n and the overland flow equation. *Trans. ASAE, 20*, 89-95.
- Richards, L. A. (1931). Capillary conduction of liquid through porous media. *Physics, 1*, 318-333.
- Rodi, W. (1984). *Turbulence models and their application in hydraulics* International Association for Hydraulic Research.
- Rose, C. W. (1993). Erosion and sedimentation. In M. Bonell, M. M. Hufschmidt & J. S. Gladwell (Eds.), *Hydrology and water management in the humid tropics: Hydrological research issues and strategies for water management* (pp. 301-343) Cambridge University Press.
- Rose, C. W., Coughlan, K. J., Ciesolka, L. A. A., & Fentie, B. (1997) Program GUEST (griffith university erosion system template), a new soil conservation methodology and application to cropping systems in tropical steeplands. *ACIAR Technical Reports, 40*, 34-58.
- Rubey, W. W. (1933). Settling velocities of gravel, sand and silt particles. *American Journal of Science, 25*(148), 325-338.
- Schmitz, G. R., Haverkamp, R., & Palacios-Velez, O. (1985). A coupled surface-subsurface model for shallow water flow over initially dry soil. *21st IAHR Congress, Int. Assoc. Hydraul. Res.* Melbourne, Victoria, Australia.
- Shoucri, M. (2006). Numerical solution of the shallow water equations with a fractional step method. *Computer Physics Communication, 176*, 23-32.
- Shu, C. W. (2009). High order weighted essentially nonoscillatory schemes for convection dominated problems. *SIAM Rev., 51*, 8222---126.
- Signals and signposts; shell energy scenarios to 2050*(2011). Royal Dutch Shell plc.
- Singh, P., & Singh, V. P. (2001). *Snow and glacier hydrology (water science and technology library)*. Boston: Kluwer Academic Publishers.
- Smerdon, E. T., & Beasley, R. P. (1961). Critical tractive forces in cohesive soils. *Agric. Eng. 42*(1), 26-29.

- Smith, G. D. (1985). *Numerical solution of partial differential equations: Finite difference methods* (3rd ed.) Oxford University Press.
- Smith, R. E., & Parlange, J. Y. (1978). A parameter-efficient hydrologic infiltration model. *Water Resources Research*, 14(3)
- Stanoyevitch, A. (2005). *Introduction to numerical ordinary and partial differential equations using MATLAB*. Hoboken, N.J.: John Wiley and Sons.
- Stoker, J. J. (1957). *Water waves*. New York, NY: Wiley Interscience.
- Strang, G. (1986). *Introduction to applied mathematics*. Wellesley, MA: Wellesley-Cambridge Press.
- Strelkoff, T. (1970). Numerical solution of saint-venant equations. *J. Hydraul. Div. Am. Soc. Civ. Eng., HY1*, 224-252.
- Swartzendruber, D., & Youngs, E. G. (1974) A comparison of physically-based infiltration equations. *Soil Science*, 117, 165-167.
- Sweeten, J. M., Garton, J. E., & Mink, A. L. (1969). Hydraulic roughness of an irrigation channel with decreasing spatially varied flow. *Trans. ASAE*, 12, 466-470.
- Sylves, R. T., & Comfort, L. K. (2012). The Exxon valdez and BP deepwater horizon oil spills: Reducing risk in socio-technical systems. *American Behavioral Scientist*, 56(76)
- Szymkiewicz, R. (2010). *Numerical modeling in open channel hydraulics* Springer.
- Takken, I., Beuselinck, L., Nachtergaele, J., Govers, G., Poesen, J., & Degraer, G. (1999). Spatial evaluation of a physically-based distributed erosion model (LISEM). *Catena*, 37(3-4), 431-447.
- Tamazian, A., Chousa, J. P., & Vadlamannati, K. C. (2009). Does higher economic and financial development lead to environmental degradation: Evidence from BRIC countries. *Energy Policy*, 37(1), 246-253.
- Tang, X., Luo, Y., Lv, J., & Wei, C. (2012). Mechanisms of soil aggregates stability in purple paddy soil under conservation tillage of Sichuan basin, China. *IFIP Advances in Information and Communication Technology*, 368, 355-370.
- Tayfur, G. (2001). Modeling two-dimensional erosion process over infiltrating surfaces. *Journal of Hydraulic Engineering*, 6(3), 259-262.
- Tchamen, G.W. & Kahawita, R.A. (1998). Modelling wetting and drying effects over complex topography. *Hydrol. Process*, 12, 1151-1182.

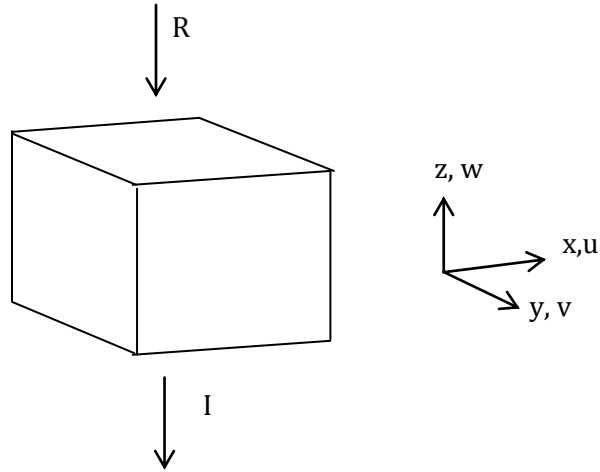
- Thomas, S., & Dawe, R. A. (2002). Review of ways to transport natural gas energy from countries which do not need the gas for domestic use. *Energy*, 28(14), 1461-1477.
- Tsanis, I. K., & Boyle, S. (2001). A 2D hydrodynamic/pollutant transport GIS model. *Advances in Engineering Software*, 32, 353-361.
- Turner, A. K., Langford, K. J., Win, M., & Clift, t. R. (1978). Discharge-depth equation for shallow flow. *J. Irrig. Drain. Div. Am. Soc. Civ. Eng.*, 104, 95-110.
- USEPA. (1994). *SWRRBWQ window's interface users guide* US Environmental Protection Agency.
- Vargaftik, N. B. (1975). *Handbook of physical properties of liquids and gases - pure substances and mixtures* (2nd ed.). New York, NY: Harper and Row Publishers Inc.
- Veil, J. A., Puder, M. G., Elcock, D., & Redweik, R. J. J. (2004). *Produced water from production of crude oil, natural gas, and coal bed methane* (White Paper No. W-31-109-Eng-38) National Energy Technology Laboratory.
- Verboom, G. K., Stelling, G. S., & Officier, M. J. (1982). Boundary conditions for the shallow water equations. In M. B. Abbott, & J. A. Cunge (Eds.), *Engineering applications of computational hydraulics* (). Boston: Pitman.
- Vertessey, R. A., Watson, F. G. R., Rahman, J. M., Cuddy, S. D., Seaton, S. P., & Chiew, F. H. (2001). New software to aid water quality management in the catchments and waterways of the south-east Queensland region. *Proceedings of the Third Australian Stream Management Conference*, pp. 611-616.
- Viney, N. R., & Sivapalan, M. (1999). A conceptual model of sediment transport: Application to the avon river basin in western Australia. *Hydrological Processes*, 13, 727-743.
- von Neumann, J. (1951). Design of computers, theory of automata and numerical analysis. In A. H. Taub (Ed.), *Collected works of J. von neumann* () Pergamon Press.
- von Neumann, J. & Richtmyer, R. D. (1950). A method for the numerical calculation of hydrodynamic shocks. *J. Comput. Phys.*, 21, 232-237.
- Vreugdenhil, C. B. (1989). *Computational hydraulics: An introduction*. Germany: Springer-Verlag.
- Watson, F., Rahman, J., & Seaton, S. (2001). Deploying environmental software using the tarsier modelling framework. *Proceedings of the Third Australian Stream Management Conference*, pp. 638.

- Wheater, H. S., Jakeman, A. J., & Beven, K. J. (1993). Progress and directions in rainfall-runoff modelling. In A. J. Jakeman, M. B. Beck & M. J. McAleer (Eds.), *Modelling change in environmental systems* (pp. 101-132). Chichester: John Wiley and Sons.
- Williams, G. P. (1970). Manning formula - A misnomer? *J. Hydraul. Div. Am. Soc. Civ. Eng.*, 96, 1930-2000.
- Williamson, K. D. (2012). The truth about fracking. *National Review*, 64(3), 26.
- Wischmeier, W. H., & Smith, D. D. (1978). *Predicting soil erosion losses: A guide to conservation planning, USDA agricultural handbook no. 537.*
- Woolhiser, D. A. (1974). Unsteady free-surface flow problems. *Institute on Unsteady Flow in Open Channels*, Colorado State University, Fort Collins. pp. 195-213.
- Woolhiser, D. A., & Liggett, J. A. (1967). Unsteady one-dimensional flow over a plane—the rising hydrograph. *Water Resources Research*, 3(3), 753-771.
- Wu, W. (2008). Review of computational river dynamics. *Journal of Hydraulic Engineering*, 134(10), 1542.
- Yakimiv, E., & Robert, A. (1986). Fractional step methods. *Monthly Weather Review*, 114, 240.
- Yalin, Y. S. (1963). An expression for bed-load transportation. *J. Hydraul. Div. Am. Soc. Civ. Eng.*, 89(HY3), 221-250.
- Yanenko, N.N. (1971). *The method of fractional steps*. New York, Berlin, Heidelberg: Springer.
- Yost, S. A., & Rao, P. M. S. V. (1998). A non-oscillatory scheme for open channel flows. *Advances in Water Resources*, 22(2), 133-143.
- Young, R. A., Onstad, C. A., Bosch, D. D., & Anderson, W. P. (1987). AGNPS, agricultural nonpoint source pollution. A watershed analysis tool. *Conservation research report 35* (). Washington, DC: US Department of Agriculture.
- Youngs, E. (1995). The physics of infiltration. *Soil Sci. Soc. Am. J.*, 59, 307-313.
- Yu, B., Rose, C. W., Cielsiolka, C. A. A., Coughlan, K. J., & Fentie, B. (1997). Towards a framework for runoff and soil loss prediction using GUEST technology. *Australian Journal of Soil Research*, 35, 1191-1212.
- Zhang, W. & Cundy, T. (1989). Modeling of two-dimensional overland flow. *Water Resources Research*, 25(9), 2019-2035.

Appendix A

Derivation of the St. Venant Equations

Consider the unit cube system



Continuity Equation (from Navier-Stokes)

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad \text{A.1}$$

Kinematic boundary conditions

$$\frac{\partial z_2}{\partial t} = w z_2 = \frac{\partial z_2}{\partial t} + u z_2 \frac{\partial z_2}{\partial x} + v z_2 \frac{\partial z_2}{\partial y} - r \quad \text{A.2}$$

$$\frac{\partial z_1}{\partial t} = w z_1 = \frac{\partial z_1}{\partial t} + u z_1 \frac{\partial z_1}{\partial x} + v z_1 \frac{\partial z_1}{\partial y} - I \quad \text{A.3}$$

Integrate the continuity equations over depth, term by term

$$\frac{1}{z_2 - z_1} \int_{z_1}^{z_2} \frac{\partial u}{\partial x} dz = \frac{1}{z_2 - z_1} \frac{\partial u}{\partial x} z \Big|_{z_1}^{z_2} = \frac{\partial u}{\partial x} \quad \text{A.4}$$

$$\frac{1}{z_2 - z_1} \int_{z_1}^{z_2} \frac{\partial v}{\partial y} dz = \frac{1}{z_2 - z_1} \frac{\partial v}{\partial y} z \Big|_{z_1}^{z_2} = \frac{\partial v}{\partial y} \quad \text{A.5}$$

For the third term, we need the Kinematic boundary conditions

$$\frac{1}{z_2 - z_1} \left[\left(\frac{\partial z_2}{\partial t} - \frac{\partial z_1}{\partial t} \right) + \left(u_{z_2} \frac{\partial z_2}{\partial x} - u_{z_1} \frac{\partial z_1}{\partial x} \right) + \left(v_{z_2} \frac{\partial z_2}{\partial y} - v_{z_1} \frac{\partial z_1}{\partial y} \right) - r + I \right] \quad \text{A.6}$$

We regroup the equation, by letting $h = z_2 - z_1$ with velocity constant with depth

$$\frac{\partial h}{\partial t} + \left(h \frac{\partial u}{\partial x} + u \frac{\partial h}{\partial x} \right) + \left(h \frac{\partial v}{\partial y} + v \frac{\partial h}{\partial y} \right) = r - I \quad \text{A.7}$$

$$\frac{\partial h}{\partial t} + \frac{\partial(hu)}{\partial x} + \frac{\partial(hv)}{\partial y} = r - I \quad \text{A.8}$$

Momentum Equations (Navier- Stokes x-direction)

$$\rho \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = - \frac{\partial p}{\partial x} + \mu \nabla^2 u + F_x \quad \text{A.9}$$

Performing a term by term integration

First:

$$\frac{1}{z_2 - z_1} \rho \int_{z_1}^{z_2} \frac{\partial u}{\partial t} dz = \rho \frac{\partial u}{\partial t} \quad \text{A.10}$$

Second:

$$\frac{1}{z_2 - z_1} \rho \int_{z_1}^{z_2} \frac{\partial u^2}{\partial x} dz = \rho \frac{\partial u^2}{\partial x} \quad \text{A.11}$$

Third:

$$\frac{1}{z_2 - z_1} \rho \int_{z_1}^{z_2} \frac{\partial uv}{\partial y} dz = \rho \frac{\partial uv}{\partial y} \quad A.12$$

Fourth: we employ the kinematic boundary conditions

$$\frac{1}{z_2 - z_1} \rho \int_{z_1}^{z_2} \frac{\partial uw}{\partial z} dz = \frac{1}{z_2 - z_1} \rho uw \Big|_{z_1}^{z_2} \quad A.13$$

$$\frac{1}{z_2 - z_1} u \left[\left(\frac{\partial z_2}{\partial t} + uz_2 \frac{\partial z_2}{\partial x} + vz_2 \frac{\partial z_2}{\partial y} - r \right) - \left(\frac{\partial z_1}{\partial t} + uz_1 \frac{\partial z_1}{\partial x} + vz_1 \frac{\partial z_1}{\partial y} - l \right) \right] \quad A.14$$

The left side of the momentum equation becomes

$$\frac{\rho}{h} \left(\frac{\partial hu}{\partial t} + \frac{\partial hu^2}{\partial x} + \frac{\partial huv}{\partial y} - ur + uf \right) \quad A.15$$

In terms of shear stress, the right side is written as

$$-\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + F_x \quad A.16$$

Assume horizontal shear components are small

$$-\frac{\partial p}{\partial x} + \frac{\partial \tau_{zx}}{\partial z} + F_x \quad A.17$$

The first term is the unbalanced pressure force; when vertically averaged:

$$\frac{1}{h} \left(-\rho gh \frac{\partial h}{\partial x} \right) \quad A.18$$

The second term must be vertically integrated:

$$\frac{1}{z_2 - z_1} \int_{z_1}^{z_2} \frac{\partial \tau_{zx}}{\partial z} dz = \frac{1}{h} \tau_{zx} \Big|_{z_1}^{z_2} = -\tau_x^{z_1} \quad A.19$$

Shear stress at the water surface is zero

The third term is the gravitational force:

$$\frac{1}{h} \left(-\rho g h \frac{\partial z_1}{\partial x} \right) \quad A.20$$

Combining and multiplying by depth

$$-\rho g h \left(\frac{\partial h}{\partial x} + \frac{\partial z_1}{\partial x} \right) - \tau_x^{z_1} \quad A.21$$

Combining all terms, the x-direction momentum equation for overland flow is

$$\rho \left(\frac{\partial hu}{\partial t} + \frac{\partial hu^2}{\partial x} + \frac{\partial huv}{\partial y} \right) = -\rho g h \left(\frac{\partial h}{\partial x} + \frac{\partial z_1}{\partial x} \right) - \tau_x^{z_1} \quad A.22$$

Similarly, the y-direction equation is

$$\rho \left(\frac{\partial hu}{\partial t} + \frac{\partial huv}{\partial x} + \frac{\partial hv^2}{\partial y} \right) = -\rho g h \left(\frac{\partial h}{\partial y} + \frac{\partial z_1}{\partial y} \right) - \tau_y^{z_1} \quad A.23$$

With some substitutions:

$$ql = r - I$$

$$S_{fx} = \tau_x^{z_1} / \rho g$$

$$S_{fy} = \tau_y^{z_1} / \rho g$$

$$S_{ox} = \partial z_1 / \partial x$$

$$S_{oy} = -\partial z_1 / \partial y$$

The resulting equations may be written in the conservation form as follows:

Conservation of mass equation

$$\frac{\partial H}{\partial t} + \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = R - I \quad A. 24$$

Conservation of momentum equation in x-direction

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = E_x \quad A. 25$$

Conservation of momentum equation in y-direction

$$\frac{\partial V}{\partial t} + \frac{\partial G}{\partial x} + \frac{\partial S}{\partial y} = E_y \quad A. 26$$

Here x and y are horizontal co-ordinates, t is time and the conservation variables are

$$H = h(x, y, t) \quad U = u(x, y, t)h(x, y, t) \quad V = v(x, y, t)h(x, y, t) \quad A. 27$$

$$F = u^2 h + \frac{1}{2} g h^2 \quad A. 28$$

$$G = uvh \quad A. 29$$

$$S = v^2 h + \frac{1}{2} g h^2 \quad A. 30$$

E_x and E_y are source terms defined as

$$E_x(x, y, t) = gH(S_{ox} - S_{fx}) \quad A. 31$$

$$E_y(x, y, t) = gH(S_{oy} - S_{fy}) \quad A. 32$$

Appendix B

Derivation of the advection-diffusion equation describing pollutant transport

Many physical problems involve simultaneous combinations of convection, diffusion, and dispersion. Advection is used to model transport of the conserved quantity through the domain by the action of some velocity field i.e. the bulk un-deformed transport of energy through a domain as a result of a pressure or head gradient. Diffusion on the other hand, is the phenomenon whereby energy spreads out spatially as time increases. It is a process whereby energy flows from places of high density to places of lower density, and the rate of energy flux is proportional to the spatial gradient of the energy density.

The rate of pollutant transport that occurs by advection is given by the product of the solute concentration c and the components of the fluid velocity while the rate of pollutant transport that occurs by diffusion is given by Fick's Law.

Transport of a solute by advection can be described by the equation:

$$\frac{\partial c}{\partial t} = u \frac{\partial c}{\partial x} + v \frac{\partial c}{\partial y} + w \frac{\partial c}{\partial z} \quad B.1a$$

which is equivalent to $\mathbf{u} \cdot \nabla c$ B.1b

where $\mathbf{u} = u, v, w$ represents the velocity in vector form and

c = concentration of quantity that is being advected

Fick's Law of diffusion is stated as:

$$\frac{\partial c}{\partial t} = D \left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} + \frac{\partial^2 c}{\partial z^2} \right) \quad B.2a$$

which is equivalent to $\quad = \mathbf{D} \nabla^2 c \quad B.2b$

where D represents the Diffusion coefficient.

When modeling phenomena in which both advection and diffusion occur, for example, a chemical that is being advected by the bulk motion of the fluid in which it is dissolved and is also diffusing through the fluid, according to Fick's law, we obtain a new constitutive law for the process simply by adding together the advection constitutive law and the diffusion constitutive law.

The advection-diffusion constitutive law is thus:

$$\mathbf{u} \cdot \nabla c + D \nabla^2 c \quad B.3$$

Since this constitutive law represents the flux of the quantity, substituting for flux in the continuity equation yields the advection-diffusion equation which describes solute or pollutant transport of a quantity through a fluid.

$$\frac{\partial c}{\partial t} = \left(u \frac{\partial c}{\partial x} + v \frac{\partial c}{\partial y} + w \frac{\partial c}{\partial z} \right) + D \left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2} + \frac{\partial^2 c}{\partial z^2} \right) + q_c \quad B.4a$$

$$\frac{\partial c}{\partial t} = \mathbf{u} \cdot \nabla c + D \nabla^2 c + q_c \quad B.4b$$

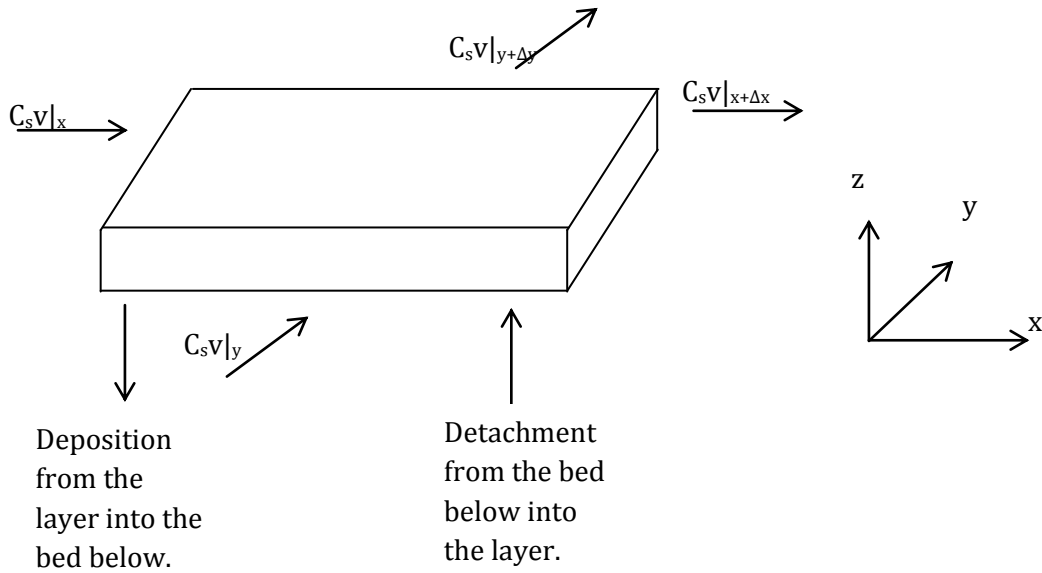
Where q_c is the pollutant source/sink term.

Appendix C

Derivation of the Sediment Transport/Erosion Model

The model is an extension of Foster's equation – $\frac{dG}{dx} = D_f$ – in 2-D

System: The fluid above the erodible bed



There can be no net accumulation in the fluid layer so

$$0 = \Delta x \Delta h (C_s v|_y - C_s v|_{y+\Delta y}) + \Delta y \Delta h (C_s u|_x - C_s u|_{x+\Delta x}) + D \Delta x \Delta y - E \Delta x \Delta y$$

C. 1

$$0 = -\frac{\partial(hu C_s)}{\partial x} - \frac{\partial(hv C_s)}{\partial y} + D - E$$

C. 2

Where E and D have units of $\frac{kg}{m^2 \cdot s}$ ($\frac{mass}{unit\ area \cdot unit\ time}$)

And C_s is the sediment concentration in mass of sand (kg)/ m^3 of fluid

or

$$\frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y} = D_f \quad C.3$$

Where

$$G_x = huC_s \quad G_y = hvC_sE \quad C.4$$

And $D_f = \text{net detachment rate}$

Which is positive for detachment and negative for deposition

Appendix D

An Overview of Classes of PDEs

Physical problems are governed by many different partial differential equations. The classification of partial differential equations was formally developed by utilizing the theory of characteristics as outlined by Crandall. The development provides considerable insight into the theoretical foundations of the classification. Partial differential equations are classified as parabolic, hyperbolic or elliptic based on the properties of the equations.

Hyperbolic Equations

Consider the generalized second order differential equation in the function u for two independent variables x and y stated below.

$$a(x, y) \frac{\partial^2 u}{\partial x^2} + b(x, y) \frac{\partial^2 u}{\partial x \partial y} + c(x, y) \frac{\partial^2 u}{\partial y^2} + f\left(x, y, u, \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right) = 0$$

The equation is classified as hyperbolic in a region B if the discriminant $b^2 - 4ac$ is positive everywhere within the region. Equations of this type require both initial and boundary conditions. The boundary conditions may consist of the value of the function (Dirichlet type), its normal derivative (Neuman type), or a combination of the function and its normal derivative (mixed type) on the region of definition. The initial conditions are the values of the function u and its first time derivative defined at some time t_0 . An example of this is the one dimensional wave equation.

$$a^2 \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial t^2}$$

Parabolic Equations

We consider again the generalized second order differential equation stated in the hyperbolic section. A parabolic partial differential equation is characterized by a zero discriminant at all points within the region B over which the equation is defined i.e. $b^2 - 4ac = 0$. Initial and boundary values are also required for a properly posed problem and just as in the hyperbolic case; the boundary conditions can be the function value, its normal derivative or a linear combination of the function and its normal derivative. An important type of a parabolic equation is the one-dimensional heat flow equation as well as the potential flow equation in saturated porous media.

Elliptic Equations

For the case of an elliptic partial differential equation, the discriminant of the generalized second order partial differential equation described in the preceding sections over a region B is negative. In contrast to parabolic and hyperbolic equations which require an open domain, elliptic equations require boundary conditions specified over a closed boundary of the region B . It requires that a value of the function, its normal derivative or a combination of both be specified to assure a unique solution. An important equation of this form is the Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y)$$

An equivalent expression of this where the source term goes to zero is the Laplace equation which describes the steady state temperature profile in two dimensional space.

Appendix E

Computer Programs

E.1 Program that generated Figures 4.2 (b) and (c)

```
%{
This script file is going to run the analytical solutions to the Steady
State St. Venant equations. The solutions were got from a Numerical
Analysis Report (3/95) from the University of Reading, UK. Title of the
Report is "Steady Open Channel Test Problems with Analytic Solutions" by I
MacDonald, M J Baines, N K Nichols and P G Samuels.
%}
% To make sure the previous run is not interfering with this run:
clear all
close all
clc

% Define the channel reach
x = linspace(0,1000,1001); % x is the distance along the channel.
g = 10; % acceleration due to gravity
Fluid_Depth = (4/g)^(1/3).*(1 + 0.5*exp(-16*(0.001* x - 0.5).^2)); % this is the
hypothetical depth profile
hold on
plot(x, Fluid_Depth,'k')
yprimex = -(4/g)^(1/3).*(2/125)*(0.001* x - 0.5).*exp(-16*(0.001* x - 0.5).^2);

% Slope function as given in the report
Slope = (1 - 4./(g*Fluid_Depth.^3)).*yprimex + 0.36*((2* Fluid_Depth+ 10).^(4/3)./(10*
Fluid_Depth).^(10/3));
plot(x, 100*Slope, 'k-')

%As given in the example:
Q = 20; % Q is the discharge
T = 10; % Channel width

Critical_Depth = (Q^2/(g*T^2))^(1/3); % The critical depth occurs when Froude number =
1
Critical_Depth = Critical_Depth*ones(1,length(x)); % turning it into an array
plot (x, Critical_Depth, 'k-')

hold off
```



```

% To get the bed profile, you integrate the slope of the bed:
% Slope = -dz/dx

Bed_Level = zeros(1,length(x)); % preallocating for speed. Z is the bed level
Bed_Level(1,length(x)) = 0; %This is an arbitrary boundary condition.
% I could put whatever value here instead of zero and it wont make a difference in the
argument
Grid_size = x(3) - x(2); % I could have used any two successive elements in x. The idea is to
get the grid size
for i = length(x):-1:2
    Bed_Level(1,i-1) = Bed_Level(1,i)+ Slope(1,i)*Grid_size;
end

figure
%This creates a new figure window. I don't want to have the two plots on
%the same figure because it is much better to display in Microsoft Word
%this way. I will add the legend and axis label in Word also

hold on
plot (x,Bed_Level, 'k', 'LineWidth', 2)

Free_Surface = Bed_Level+Fluid_Depth;

plot(x,Free_Surface,'k')

Critical_Surface = Bed_Level + Critical_Depth;

plot(x, Critical_Surface, 'k-.')

hold off

```

E.2 Program that generated Figures 4.3

```

%{
This script file calls the 1-D MacCormack function file and displays the
result. It is where the input values are set.
%}
clc
clear all
close all
tic
%% This section prepares the visualization of results
scrnsz = get(0,'ScreenSize');

```

```
figure('Position',[1 1 scrnsz(3) 0.9*scrnsz(4)])
```

```
%%
```

```
t = 0; % Initialize time
```

```
T = 800; % Simulation Time in seconds
```

```
del_t = 0.1; % Time step
```

```
grid_size = 1; % channel length chopped into 1m intervals
```

```
g = 10; % Acceleration due to gravity
```

```
%% DEM, Z, will be used here to test the code
```

```
Z =
```

```
[7.23051126373644,7.21888464542185,7.20726338179063,7.19564754326263,7.18403  
720092306,7.17243242652359,7.16083329248348,7.14923987189050,7.137652238501  
78,7.12607046674453,7.11449463171661,7.10292480918701,7.09136107559620,7.079  
80350805632,7.06825218435126,7.05670718293662,7.04516858293947,7.0336364641  
5805,7.02211090706129,7.01059199278816,6.99907980314692,6.98757442061420,6.9  
7607592833389,6.96458441011597,6.95309995043509,6.94162263442902,6.93015254  
789697,6.91868977729769,6.90723440974747,6.89578653301792,6.88434623553358,6  
.87291360636940,6.86148873524801,6.85007171253682,6.83866262924491,6.8272615  
7701984,6.81586864814414,6.80448393553172,6.79310753272405,6.78173953388611,  
6.77038003380227,6.75902912787183,6.74768691210447,6.73635348311547,6.725028  
93812070,6.71371337493146,6.70240689194910,6.69110958815941,6.6798215631268  
4,6.66854291698851,6.65727375044799,6.64601416476888,6.63476426176819,6.6235  
2414380949,6.61229391379588,6.60107367516270,6.58986353187003,6.57866358839  
505,6.56747394972406,6.55629472134439,6.54512600923603,6.53396791986305,6.52  
282056016483,6.51168403754702,6.50055845987234,6.48944393545111,6.478340573  
03156,6.46724848178996,6.45616777132051,6.44509855162494,6.43404093310205,6.  
42299502653684,6.41196094308956,6.40093879428448,6.38992869199845,6.3789307  
4844922,6.36794507618362,6.35697178806541,6.34601099726302,6.33506281723699,  
6.32412736172727,6.31320474474024,6.30229508053557,6.29139848361288,6.280515  
06869811,6.26964495072980,6.25878824484507,6.24794506636545,6.2371155307825  
3,6.22629975374331,6.21549785103553,6.20470993857262,6.19393613237860,6.1831  
7654857271,6.17243130335394,6.16170051298528,6.15098429377789,6.14028276207  
500,6.12959603423576,6.11892422661880,6.10826745556574,6.09762583738442,6.08  
699948833210,6.07638852459846,6.06579306228838,6.05521321740473,6.044649105  
83089,6.03410084331321,6.02356854544330,6.01305232764026,6.00255230513271,5.  
99206859294078,5.98160130585794,5.97115055843278,5.96071646495065,5.9502991  
3941524,5.93989869553004,5.92951524667977,5.91914890591167,5.90879978591685,  
5.89846799901139,5.88815365711757,5.87785687174493,5.86757775397136,5.857316  
41442412,5.84707296326082,5.83684751015043,5.82664016425424,5.8164510342067  
8,5.80628022809685,5.79612785344841,5.78599401720162,5.77587882569380,5.7657  
8238464047,5.75570479911644,5.74564617353688,5.73560661163850,5.72558621646  
078,5.71558509032723,5.70560333482679,5.69564105079520,5.68569833829661,5.67  
577529660515,5.66587202418667,5.65598861868063,5.64612517688200,5.636281794  
72337,5.62645856725722,5.61665558863824,5.60687295210586,5.59711074996698,5.  
58736907357876,5.57764801333168,5.56794765863272,5.55826809788879,5.5486094  
1849030,5.53897170679498,5.52935504811189,5.51975952668561,5.51018522568080,
```

5.50063222716679,5.49110061210260,5.48159046032209,5.47210185051944,5.462634
86023484,5.45318956584048,5.44376604252680,5.43436436428910,5.4249846039142
9,5.41562683296809,5.40629112178249,5.39697753944343,5.38768615377896,5.3784
1703134758,5.36917023742700,5.35994583600317,5.35074388975972,5.34156446006
770,5.33240760697568,5.32327338920020,5.31416186411662,5.30507308775029,5.29
600711476814,5.28696399847057,5.27794379078383,5.26894654225270,5.259972302
03358,5.25102111788799,5.24209303617644,5.23318810185276,5.22430635845873,5.
21544784811922,5.20661261153767,5.19780068799202,5.18901211533105,5.1802469
2997110,5.17150516689325,5.16278685964091,5.15409204031782,5.14542073958647,
5.13677298666699,5.12814880933636,5.11954823392819,5.11097128533277,5.102417
98699771,5.09388836092883,5.08538242769162,5.07690020641306,5.0684417147838
5,5.06000696906112,5.05159598407149,5.04320877321460,5.03484534846705,5.0265
0572038675,5.01818989811768,5.00989788939506,5.00162970055094,4.99338533652
020,4.98516480084694,4.97696809569125,4.96879522183645,4.96064617869662,4.95
252096432459,4.94441957542032,4.93634200733959,4.92828825410316,4.920258308
40618,4.91225216162813,4.90426980384294,4.89631122382960,4.88837640908305,4.
88046534582545,4.87257801901775,4.86471441237164,4.85687450836176,4.8490582
8823830,4.84126573203987,4.83349681860667,4.82575152559398,4.81802982948595,
4.81033170560961,4.80265712814926,4.79500607016102,4.78737850358773,4.779774
39927405,4.77219372698182,4.76463645540565,4.75710255218878,4.7495919839391
0,4.74210471624543,4.73464071369395,4.72719993988492,4.71978235744948,4.7123
8792806670,4.70501661248077,4.69766837051836,4.69034316110614,4.68304094228
840,4.67576167124488,4.66850530430866,4.66127179698418,4.65406110396540,4.64
687317915400,4.63970797567773,4.63256544590882,4.62544554148241,4.618348213
31511,4.61127341162360,4.60422108594322,4.59719118514668,4.59018365746270,4.
58319845049475,4.57623551123975,4.56929478610676,4.56237622093571,4.5554797
6101607,4.54860535110546,4.54175293544832,4.53492245779444,4.52811386141748,
4.52132708913342,4.51456208331893,4.50781878592967,4.50109713851849,4.494397
08225356,4.48771855793634,4.48106150601952,4.47442586662476,4.4678115795603
2,4.46121858433860,4.45464682019352,4.44809622609765,4.44156674077935,4.4350
5830273962,4.42857085026882,4.42210432146320,4.41565865424124,4.40923378635
985,4.40282965543026,4.39644619893382,4.39008335423751,4.38374105860927,4.37
741924923309,4.37111786322387,4.36483683764205,4.35857610950798,4.352335615
81610,4.34611529354877,4.33991507968997,4.33373491123858,4.32757472522156,4.
32143445870675,4.31531404881539,4.30921343273448,4.30313254772869,4.2970713
3115209,4.29102972045959,4.28500765321802,4.27900506711697,4.27302189997932,
4.26705808977142,4.26111357461304,4.25518829278691,4.24928218274808,4.243395
18313283,4.23752723276735,4.23167827067606,4.22584823608963,4.2200370684527
0,4.21424470743118,4.20847109291936,4.20271616504659,4.19697986418364,4.1912
6213094884,4.18556290621372,4.17988213110846,4.17421974702693,4.16857569563
144,4.16294991885718,4.15734235891620,4.15175295830126,4.14618165978918,4.14
062840644394,4.13509314161943,4.12957580896187,4.12407635241194,4.118594716
20651,4.11313084488007,4.10768468326593,4.10225617649691,4.09684527000590,4.
09145190952597,4.08607604109021,4.08071761103126,4.07537656598051,4.0700528
5286698,4.06474641891593,4.05945721164711,4.05418517887271,4.04893026869510,
4.04369242950410,4.03847160997412,4.03326775906089,4.02808082599796,4.022910

76029291,4.01775751172321,4.01262103033189,4.00750126642284,4.0023981705559
3,3.99731169354179,3.99224178643629,3.98718840053489,3.98215148736657,3.9771
3099868761,3.97212688647505,3.96713910291998,3.96216760042049,3.95721233157
442,3.95227324917192,3.94735030618766,3.94244345577297,3.93755265124756,3.93
267784609122,3.92781899393514,3.92297604855312,3.91814896385251,3.913337693
86501,3.90854219273720,3.90376241472091,3.89899831416342,3.89424984549743,3.
88951696323090,3.88479962193665,3.88009777624187,3.87541138081736,3.8707403
9036674,3.86608475961535,3.86144444329913,3.85681939615328,3.85220957290081,
3.84761492824090,3.84303541683721,3.83847099330598,3.83392161220407,3.829387
22801681,3.82486779514584,3.82036326789672,3.81587360046654,3.8113987469313
5,3.80693866123355,3.80249329716917,3.79806260837508,3.79364654831604,3.7892
4507027185,3.78485812732421,3.78048567234370,3.77612765797658,3.77178403663
158,3.76745476046668,3.76313978137573,3.75883905097514,3.75455252059047,3.75
028014124301,3.74602186363634,3.74177763814282,3.73754741479015,3.733331143
24778,3.72912877281347,3.72494025239973,3.72076553052026,3.71660455527651,3.
71245727434409,3.70832363495930,3.70420358390565,3.70009706750037,3.6960040
3158101,3.69192442149201,3.68785818207132,3.68380525763705,3.67976559197425,
3.67573912832155,3.67172580935806,3.66772557719018,3.66373837333851,3.659764
13872489,3.65580281365934,3.65185433782729,3.64791865027668,3.6439956894052
9,3.64008539294805,3.63618769796450,3.63230254082631,3.62842985720488,3.6245
6958205908,3.62072164962306,3.61688599339414,3.61306254612090,3.60925123979
123,3.60545200562065,3.60166477404063,3.59788947468713,3.59412603638915,3.59
037438715751,3.58663445417370,3.58290616377890,3.57918944146309,3.575484211
85436,3.57179039870828,3.56810792489753,3.56443671240155,3.56077668229644,3.
55712775474498,3.55348984898676,3.54986288332857,3.54624677513484,3.5426414
4081833,3.53904679583095,3.53546275465474,3.53188923079305,3.52832613676188,
3.52477338408138,3.52123088326753,3.51769854382408,3.51417627423453,3.510663
98195443,3.50716157340377,3.50366895395963,3.50018602794898,3.4967126986416
8,3.49324886824370,3.48979443789049,3.48634930764058,3.48291337646941,3.4794
8654226329,3.47606870181363,3.47265975081135,3.46925958384146,3.46586809437
795,3.46248517477876,3.45911071628109,3.45574460899683,3.45238674190823,3.44
903700286381,3.44569527857448,3.44236145460984,3.43903541539477,3.435717044
20616,3.43240622316993,3.42910283325824,3.42580675428692,3.42251786491317,3.
41923604263340,3.41596116378140,3.41269310352664,3.40943173587292,3.4061769
3365707,3.40292856854810,3.39968651104637,3.39645063048318,3.39322079502041,
3.38999687165057,3.38677872619695,3.38356622331406,3.38035922648831,3.377157
59803889,3.37396119911896,3.37076988971697,3.36758352865829,3.3644019736070
5,3.36122508106827,3.35805270639011,3.35488470376649,3.35172092623984,3.3485
6122570416,3.34540545290831,3.34225345745946,3.33910508782692,3.33596019134
603,3.33281861422247,3.32968020153667,3.32654479724852,3.32341224420232,3.32
028238413192,3.31715505766619,3.31403010433459,3.31090736257313,3.307786669
73043,3.30466786207412,3.30155077479742,3.29843524202595,3.29532109682483,3.
29220817120594,3.28909629613549,3.28598530154175,3.28287501632309,3.2797652
6835618,3.27665588450446,3.27354669062686,3.27043751158669,3.26732817126085,
3.26421849254916,3.26110829738402,3.25799740674024,3.25488564064511,3.251772
81818871,3.24865875753441,3.24554327592968,3.24242618971699,3.2393073143450

7,3.23618646438030,3.23306345351836,3.22993809459610,3.22681019960362,3.2236
7957969658,3.22054604520872,3.21740940566457,3.21426946979244,3.21112604553
759,3.20797894007557,3.20482795982585,3.20167291046561,3.19851359694377,3.19
534982349518,3.19218139365510,3.18900811027377,3.18582977553133,3.182646190
95279,3.17945715742335,3.17626247520376,3.17306194394606,3.16985536270936,3.
16664252997589,3.16342324366727,3.16019730116089,3.15696449930658,3.1537246
3444339,3.15047750241657,3.14722289859481,3.14396061788756,3.14069045476262,
3.13741220326384,3.13412565702909,3.13083060930827,3.12752685298165,3.124214
18057831,3.12089238429470,3.11756125601347,3.11422058732243,3.1108701695336
3,3.10750979370269,3.10413925064822,3.10075833097142,3.09736682507585,3.0939
6452318738,3.09055121537420,3.08712669156711,3.08369074157984,3.08024315512
960,3.07678372185771,3.07331223135044,3.06982847315993,3.06633223682527,3.06
282331189371,3.05930148794199,3.05576655459784,3.05221830156156,3.048656518
62773,3.04508099570706,3.04149152284834,3.03788789026049,3.03426988833480,3.
03063730766713,3.02698993908039,3.02332757364697,3.01965000271137,3.0159570
1791288,3.01224841120835,3.00852397489508,3.00478350163375,3.00102678447148,
2.99725361686494,2.99346379270353,2.98965710633266,2.98583335257705,2.981992
32676417,2.97813382474766,2.97425764293082,2.97036357829025,2.9664514283993
7,2.96252099145215,2.95857206628675,2.95460445240930,2.95061795001764,2.9466
1236002512,2.94258748408440,2.93854312461133,2.93447908480877,2.93039516869
045,2.92629118110487,2.92216692775913,2.91802221524284,2.91385685105195,2.90
967064361261,2.90546340230502,2.90123493748720,2.89698506051882,2.892713583
78491,2.88842032071962,2.88410508582985,2.87976769471889,2.87540796411004,2.
87102571187005,2.86662075703265,2.86219291982191,2.85774202167554,2.8532678
8526816,2.84877033453445,2.84424919469220,2.83970429226530,2.83513545510661,
2.83054251242072,2.82592529478664,2.82128363418028,2.81661736399693,2.811926
31907352,2.80721033571080,2.80246925169532,2.79770290632134,2.7929111404125
4,2.78809379634359,2.78325071806157,2.77838175110715,2.77348674263572,2.7685
6554143825,2.76361799796196,2.75864396433084,2.75364329436600,2.74861584360
571,2.74356146932531,2.73848003055693,2.73337138810891,2.72823540458502,2.72
307194440351,2.71788087381581,2.71266206092510,2.70741537570454,2.702140690
01526,2.69683787762418,2.69150681422140,2.68614737743747,2.68075944686029,2.
67534290405174,2.66989763256406,2.66442351795588,2.65892044780798,2.6533883
1173875,2.64782700141929,2.64223641058827,2.63661643506637,2.63096697277051,
2.62528792372761,2.61957919008812,2.61384067613918,2.60807228831738,2.602273
93522125,2.59644552762330,2.59058697848177,2.58469820295201,2.5787791183974
0,2.57282964440003,2.56684970277085,2.56083921755959,2.55479811506414,2.5487
2632383965,2.54262377470718,2.53649040076195,2.53032613738120,2.52413092223
163,2.51790469527644,2.51164739878194,2.50535897732375,2.49903937779259,2.49
268854939961,2.48630644368132,2.47989301450412,2.47344821806829,2.466972012
91169,2.46046435991293,2.45392522229408,2.44735456562305,2.44075235781539,2.
43411856913576,2.42745317219890,2.42075614197012,2.41402745576544,2.4072670
9325118,2.40047503644316,2.39365126970543,2.38679577974854,2.37990855562739,
2.37298958873856,2.36603887281731,2.35905640393398,2.35204218049007,2.344996
20321376,2.33791847515509,2.33080900168058,2.32366779046748,2.3164948514975
8,2.30929019705048,2.30205384169653,2.29478580228927,2.28748609795744,2.2801

5475009654,2.27279178235998,2.26539722064978,2.25797109310686,2.25051343010
085,2.24302426421955,2.23550363025793,2.22795156520669,2.22036810824044,2.21
275330070548,2.20510718610713,2.19742981009667,2.18972122045794,2.181981467
09344,2.17421060201014,2.16640867930484,2.15857575514917,2.15071188777423,2.
14281713745485,2.13489156649346,2.12693523920364,2.11894822189329,2.1109305
8284744,2.10288239231077,2.09480372246975,2.08669464743441,2.07855524321992,
2.07038558772768,2.06218576072621,2.05395584383172,2.04569592048838,2.037406
07594821,2.02908639725085,2.02073697320289,2.01235789435704,2.0039492529909
6,1.99551114308591,1.98704366030505,1.97854690197158,1.97002096704663,1.9614
6595610688,1.95288197132201,1.94426911643189,1.93562749672361,1.92695721900
829,1.91825839159769,1.90953112428066,1.90077552829940,1.89199171632554,1.88
317980243613,1.87433990208938,1.86547213210033,1.85657661061635,1.847653457
09254,1.83870279226697,1.82972473813589,1.82071941792870,1.81168695608299,1.
80262747821934,1.79354111111618,1.78442798268446,1.77528822194231,1.7661219
5898966,1.75692932498282,1.74771045210891,1.73846547356045,1.72919452350973,
1.71989773708331,1.71057525033642,1.70122720022741,1.69185372459219,1.682454
96211867,1.67303105232124,1.66358213551528,1.65410835279171,1.6446098459915
1,1.63508675768043,1.62553923112365,1.61596741026051,1.60637143967935,1.5967
5146459244,1.58710763081091,1.57744008471989,1.56774897325363,1.55803444387
082,1.54829664452996,1.53853572366486,1.52875183016031,1.51894511332777,1.50
911572288131,1.49926380891366,1.48938952187231,1.47949301253594,1.469574431
99081,1.45963393160749,1.44967166301763,1.43968777809094,1.42968242891240,1.
41965576775957,1.40960794708014,1.39953911946964,1.38944943764939,1.3793390
5444459,1.36920812276268,1.35905679557185,1.34888522587978,1.33869356671261,
1.32848197109416,1.31825059202527,1.30799958246347,1.29772909530284,1.287439
28335410,1.27713029932495,1.26680229580061,1.25645542522467,1.2460898398801
2,1.23570569187066,1.22530313310225,1.21488231526492,1.20444338981482,1.1939
8650795653,1.18351182062563,1.17301947847153,1.16250963184057,1.15198243075
936,1.14143802491837,1.13087656365587,1.12029819594201,1.10970307036329,1.09
909133510718,1.08846313794714,1.07781862622777,1.06715794685036,1.056481246
25858,1.04578867042457,1.03508036483522,1.02435647447874,1.01361714383148,1.
00286251684509,0.992092736933869,0.981307946962432,0.970508289233634,0.9596
93905476765,0.948864936836017,0.938021523859212,0.927163806486807,0.9162919
24041158,0.905406015216056,0.894506218066528,0.883592669998896,0.8726655077
61104,0.861724867433307,0.850770884418717,0.839803693434710,0.8288234285041
94,0.817830222947222,0.806824209372874,0.795805519671379,0.784774285006497,0.
.773730635808146,0.762674701765274,0.751606611818984,0.740526494155895,0.729
434476201749,0.718330684615255,0.707215245282175,0.696088283309634,0.684949
923020680,0.673800287949058,0.662639500834225,0.651467683616581,0.640284957
432932,0.629091442612173,0.617887258671182,0.606672524310942,0.595447357412
867,0.584211875035347,0.572966193410496,0.561710427941111,0.550444693197830,
0.539169102916495,0.527883769995709,0.516588806494595,0.505284323630740,0.49
3970431778338,0.482647240466510,0.471314858377819,0.459973393346962,0.44862
2952359640,0.437263641551605,0.425895566207888,0.414518830762184,0.40313353
8796417,0.391739793040467,0.380337695372055,0.368927346816795,0.35750884754
8395,0.346082296889022,0.334647793309805,0.323205434431502,0.31175531702529

```

9,0.300297537013760,0.288832189471916,0.277359368628486,0.265879167867241,0.
254391679728496,0.242896995910728,0.231395207272325,0.219886403833456,0.208
370674778067,0.196848108455985,0.185318792385150,0.173782813253951,0.162240
256923675,0.150691208431069,0.139135751990996,0.127573970999203,0.116005948
035187,0.104431764865156,0.0928515024450876,0.0812652409238769,0.0696730596
465781,0.0580750371577313,0.0464712512047745,0.0348617787415392,0.023246695
9318254,0.0116260781530549,0;];

```

```

%Slope = ones(1,length(Z)); % pre-allocating the Slope array for speed

```

```

for i = 1:length(Z)-1
    Slope(i) = - (Z(i+1)-Z(i))/grid_size;
end

```

```

Slope = [Slope Slope(end)]; % this ensures the pre-allocation doesn't affect the boundary
value of the Slope

```

```

%% Initial conditions of fluid depth and velocity as given in the Test Problems
h_o = 0.748409*ones(1,length(Z)); % initial flow depth (in metres)
u_o = 2/0.748409*ones(1,length(Z)); % velocity

```

```

%%
t = t + del_t;
n = 0; %start counter
while t < T
    % The MacCormack Method is used to solve the Shallow Water equations

```

```

    [h_c,u_c] = NumericalSolution(h_o,u_o,del_t,grid_size,Slope);

```

```

    %The output serve as input at the next time step
    h_o = h_c; u_o = u_c;

```

```

    % Advance one time step
    t = t + del_t;

```

```

    n = n+ 1; %increase counter by 1

```

```

%% Movie of the Simulation

```

```

    plot(h_o)
    xlim([0 1000]); ylim([0.3 1.3])
    drawnow
    pause(0.01)
end
toc

```

E.3 Program that generated Figures 4.4.2 (b) and (c)

```
%{
This script file is going to run the analytical solutions to the Steady
State St. Venant equations. The solutions were got from a Numerical
Analysis Report (3/95) from the University of Reading, UK. Title of the
Report is "Steady Open Channel Test Problems with Analytic Solutions" by I
MacDonald, M J Baines, N K Nichols and P G Samuels.
%}
% To make sure the previous run is not interfering with this run:
clear g n i t T Slope Z
close all
clc

%% Define the channel reach

x = linspace(0,1000,1001); % x is the distance along the channel.
g = 10; % acceleration due to gravity
a = [-0.348427 0.552264 -0.555580]; % Vector containing terms required for calculation of
fluid depth

%% Calculate the Fluid depth and its derivative

Fluid_Depth = zeros(1,length(x)); % preallocating for speed.
yprimex = zeros(1,length(x)); % preallocating for speed.
for i = 1:length(x)
    if x(i) >= 0 && x(i) <= 500
        Fluid_Depth(i) = (4/g)^(1/3)*(0.9 - 0.1667*exp(-0.004*x(i))); % this is the
        hypothetical depth profile
        yprimex(i) = (4/g)^(1/3)*(0.000667*exp(-0.004*x(i)));
    else
        Summation = 0; Summationprime = 0;
        for k = 1:3
            Summation = Summation + a(k)*exp(-20*k*(0.001*x(i) - 0.5));
            Summationprime = Summationprime + k*a(k)*exp(-20*k*(0.001*x(i) - 0.5));
        end
        Fluid_Depth(i) = (4/g)^(1/3)*(1 + Summation + (0.8*exp(0.001*x(i) - 1)));
        yprimex(i) = (4/g)^(1/3)*(-0.02*Summationprime + (0.0008*exp(0.001*x(i) - 1)));
    end
end
%% Plotting Section
hold on
plot(x, Fluid_Depth,'k')

% Slope function as given in the report
```



```
Slope = (1 - 4./(g*Fluid_Depth.^3)).*yprimex + 0.16*((2* Fluid_Depth+ 10).^4/3)./(10*
Fluid_Depth).^10/3);
plot(x, 50*Slope, 'k--')
```

```
%As given in the example:
Q = 20; % Q is the discharge
T = 10; % Channel width
```

```
Critical_Depth = (Q^2/(g*T^2))^(1/3); % The critical depth occurs when Froude number =
1
Critical_Depth = Critical_Depth*ones(1,length(x)); % turning it into an array
plot (x, Critical_Depth, 'k-.')
```

```
hold off
```

```
% To get the bed profile, you integrate the slope of the bed:
% Slope = -dz/dx
```

```
Bed_Level = zeros(1,length(x)); % preallocating for speed. Z is the bed level
Bed_Level(1,length(x)) = 0; %This is an arbitrary boundary condition.
% I could put whatever value here instead of zero and it wont make a difference in the
argument
Grid_size = x(3) - x(2); % I could have used any two successive elements in x. The idea is to
get the grid size
for i = length(x):-1:2
    Bed_Level(1,i-1) = Bed_Level(1,i)+ Slope(1,i)*Grid_size;
end
```

```
figure
%This creates a new figure window. I don't want to have the two plots on
%the same figure because it is much better to display in Microsoft Word
%this way. I will add the legend and axis label in Word also
```

```
hold on
plot (x,Bed_Level, 'k', 'LineWidth', 2)
```

```
Free_Surface = Bed_Level+Fluid_Depth;
```

```
plot(x,Free_Surface,'k')
```

```
Critical_Surface = Bed_Level + Critical_Depth;
```

```
plot(x, Critical_Surface, 'k-.')
```

```
hold off
```

E.3 Program that generated Figure 4.4.3 and 4.4.4

```
function [h_c,u_c] = NumericalSolution4(h_o,u_o,del_t,grid_size,Slope)
%{
This function solves the Shallow Water Equations in One-D using the
MacCormack Scheme. The results from here would be compared with the
analytical solutions.
%}

%% Boundary Conditions for the Predictor Part
h_o(1) = 0.543853; h_o(end) = 1.334899; % As given in the analytical version of the
problem
%u_o(1) = 2/0.748409; u_o(end) = 2/0.748409; % Velocity BCs, again it is as stated in the
problem

%% CONSTANTS
g = 10 ;          % acceleration due to gravity
n = 0.02;         % n is the Manning coefficient taken from Chaudhry (1993)
k = 3.7;          % is a parameter used to regulate the amount of dissipation, will be determined by
trial-and-error

Sfx = n^2*u_o.^2./h_o.^(4/3); % Friction slope calculated from the Manning equation.
Strictly speaking,
% the numerator ought to be u.*abs(u) but we know u is all positive in this example so it
doesn't matter here
U_o = u_o.*h_o; % The Unit width discharge
F_o = u_o.^2.*h_o + 0.5*g*h_o.^2; % F is a conservation variable. It is called a "conservation"
variable because the St. Venant
% equation is cast in the conservation form and it is one of those variables that makes the
momentum equation fit into a nice matrix

%% The predictor step (backward difference)
% h_p = zeros(1,length(Slope));
U_p = 2*ones(1,length(Slope));
for i = 2:length(Slope)
    h_p(i) = h_o(i) - (del_t/grid_size)*(U_o(i)-U_o(i-1));
    U_p(i) = U_o(i) - (del_t/grid_size)*(F_o(i)-F_o(i-1))+ del_t*g*0.5*(h_o(i)+h_o(i-
1)).*(Slope(i-1) - Sfx(i));
end
h_p = [0.543853 h_p(2:end)];
%U_p = [2.0 U_p(2:end)];
%h_p(1) = 0.748709; h_p(end) = 0.748709; % this removes the zero introduced at the
boundary due to pre-allocation
%U_p(1) = 2.0; U_p(end) = 2.0;
u_p = U_p./h_p; % backing out u
```

```
%% Boundary Conditions for the Corrector Part
```

```
% Calculate Intermediate Values to be used in the Corrector Part
```

```
F_p = u_p.^2.*h_p + 0.5*g*h_p.^2;  
Sfxp = n^2*u_p.^2./h_p.^(4/3);
```

```
%% The corrector step (forward difference)
```

```
% h_c = zeros(1,length(Slope));  
U_c = 2*ones(1,length(Slope));
```

```
for i = 1:length(Slope)-1
```

```
h_c(i) = 0.5 * (h_o(i) + h_p(i) - (del_t/grid_size) * (U_p(i+1)-U_p(i)));  
U_c(i) = 0.5 * (U_o(i) + U_p(i) - (del_t/grid_size) * (F_p(i+1)-F_p(i)) +  
del_t*g*0.5*(h_p(i+1)+h_p(i)).*(Slope(i+1) - Sfxp(i)));  
end
```

```
h_c = [h_c(1:end) 1.334899];
```

```
%U_c = [2.0 U_c(2:end)];
```

```
%h_c(1) = 0.748709; h_c(end) = 0.748709; % this removes the zero introduced at the  
boundary due to pre-allocation
```

```
%U_c(1) = 2.0; U_c(end) = 2.0;
```

```
u_c = U_c./h_c; %backing out u
```

```
%% Computing variables to be used in handling steep gradients (see the note in the  
"artificialviscosity" file for details):
```

```
[eps_back_x_hc, eps_for_x_hc] = artificialviscosity1D(h_c,k);
```

```
[eps_back_x_uc, eps_for_x_uc] = artificialviscosity1D(u_c,k);
```

```
% Update the INTERIOR values of h and U:
```

```
for i = 2:length(Slope)-1
```

```
h_c(i) = h_c(i) + eps_for_x_hc(i).*(h_c(i+1) - h_c(i)) - eps_back_x_hc(i).*(h_c(i) - h_c(i-1));
```

```
u_c(i) = u_c(i) + eps_for_x_uc(i).*(u_c(i+1) - u_c(i)) - eps_back_x_uc(i).*(u_c(i) - u_c(i-1));
```

```
end
```

```
% Note that since boundary conditions are "imposed," the boundary values don't need to  
be updated because we don't want them to change.
```

E.4 Function file called by Programs in E.1 to E.3

```
function[eps_back_x, eps_for_x] = artificialviscosity1D(h_o,k)
```

```
%{
```

```
ARTIFICIAL VISCOSITY COMPUTATION 1D Saint Venant Equation
```

```
Jameson, et al. 1981 developed a procedure to dampen the high frequency oscillations  
observed near steep gradients.
```

```
These oscillations are produced as a result of dispersive errors in the MacCormack  
Scheme.
```

Details about this procedure can be found in Chaudhry (1993), Chapter 8: Computation of Rapidly Varied Flows, In: "Open-Channel Flow". It is strongly recommended that this reference chapter in the book be read before attempting to use - or debug - this function file.

Please keep in mind that changes in the x-Cartesian coordinates affect columns (y) in MATLAB while changes along the y-Cartesian coordinates affect rows(x) in MATLAB. This dichotomy between the Cartesian coordinate system and how MATLAB interprets rows and columns is a common cause of confusion when computing gradients in more than one-dimension. Thankfully, this function file is for 1D cases so we don't need to be "careful" in one sense.

```

%%}

%% Pre-process Data
h_visc = h_o; % Set Initial value of testing array h_visc from input array h_o provided.
N_Cols = length(h_o);
viscos_x = zeros([1 N_Cols]);

%% Viscosity Calculation

viscos_x(1) = abs(h_visc(2)- h_visc(1))/(abs(h_visc(2))+ abs(h_visc(1)));
viscos_x(end) = abs(h_visc(end)- h_visc(end-1))/(abs(h_visc(end))+ abs(h_visc(end-1)));

for i = 2:N_Cols-1; % This loop accounts for operations carried out on all internal rows of
the matrix.
    viscos_x(i) = abs(h_visc(i+1)- 2*h_visc(i)+ h_visc(i-1))/(abs(h_visc(i+1))+
abs(2*h_visc(i))+ abs(h_visc(i-1)));
end

eps_back_x(1) = k*viscos_x(1); % This is like a boundary condition of some sort since
there's no "zeroth" element to compare the first one with

for i = 2:N_Cols;
    eps_back_x(i) = k*max(viscos_x(i-1), viscos_x(i));
end

eps_for_x(N_Cols) = k*viscos_x(N_Cols); % This is like a boundary condition of some sort
since there's no "outside" element to compare the last one with

for i = 1:N_Cols - 1;
    eps_for_x(i) = k*max(viscos_x(i), viscos_x(i+1));
end

```

E.5 Program that generated Figures 4.7 through 4.13

```
%{
This script file calls the other function files of this program. It will later be a function file
itself that will be called in some way. We make it a script file for now for debugging
purposes.
Units are MKS throughout meaning i.e. time is in seconds, lengths are in meters etc.
%}
clc
clear all
close all
tic
%% This section prepares the visualization of results
scrnsz = get(0,'ScreenSize');
figure('Position',[1 1 scrnsz(3) 0.9*scrnsz(4)],'Name','Video of Fluid
Depth','NumberTitle','off')
% h = waitbar(0,'1','Name','We keep getting better...',...
%       'CreateCancelBtn',...
%       'setappdata(gcf,'canceling',1)');
% setappdata(h,'canceling',0)

%%
t = 0;    % Initialize time
T = 40;   % Simulation Time in seconds
del_t = 0.0001; % A first estimate, the model should adapt this size as it runs
grid_size = 3; % the DEM data we have a 5m-resolution
g = 9.81;  % Acceleration due to gravity
%% DEM, Z, will be used here to test the code
%Obtained from Judsonian data Set (51:100 41:80)
%Z = ones(209,209);
%gently sloping surface
% xsc = linspace(109,9,211);
% Z = xsc(ones(1,211),:);
%Z =
[113.734,113.534,113.334,113.034,112.834,112.633,112.433,112.333,112.133,111.833,11
1.333,110.733,110.233,109.733,109.233,108.633,107.933,107.433,107.133,106.833,106.5
33,106.333,105.933,105.632,105.332,105.032,105.232,105.432,105.732,105.732,105.732,
105.632,105.532,105.432,105.132,104.832,104.732,104.732,104.932,105.132;114.134,11
3.934,113.734,113.434,113.234,113.133,112.933,112.833,112.633,112.333,111.933,111.4
33,110.933,110.433,109.933,109.033,108.133,107.633,107.133,106.833,106.533,106.333,
106.233,105.932,105.632,105.432,105.332,105.532,105.732,105.732,105.732,105.632,105
.632,105.532,105.332,105.132,104.932,104.932,104.932,105.132;114.434,114.234,114.03
4,113.834,113.634,113.533,113.333,113.233,113.133,112.833,112.533,112.133,111.733,1
11.233,110.433,108.933,107.933,107.633,107.133,106.733,106.533,106.333,106.233,106.
232,105.932,105.732,105.632,105.632,105.832,105.832,105.832,105.832,105.832,105.832
,105.632,105.332,105.232,105.132,105.132,105.132;114.734,114.634,114.434,114.234,11
```

4.034,113.833,113.733,113.633,113.533,113.433,113.133,112.833,112.333,111.633,110.433,108.833,107.833,107.533,107.133,106.733,106.533,106.433,106.333,106.333,106.232,106.032,105.932,105.932,105.932,105.932,105.832,105.832,105.932,105.932,105.832,105.532,105.332,105.332,105.332,105.332;115.134,114.934,114.734,114.534,114.334,114.133,114.033,113.933,113.933,113.933,113.833,113.333,112.633,111.333,110.133,108.933,107.933,107.433,107.133,106.833,106.633,106.633,106.533,106.333,106.332,106.232,106.132,106.132,106.132,106.032,106.032,105.932,105.932,106.032,105.932,105.632,105.432,105.332,105.332,105.432;115.434,115.234,115.034,114.734,114.534,114.334,114.233,114.133,114.133,114.333,114.433,113.733,112.733,111.433,109.933,108.733,107.933,107.533,107.133,106.833,106.633,106.633,106.633,106.533,106.432,106.432,106.432,106.432,106.432,106.332,106.232,106.132,106.032,106.132,106.032,105.832,105.632,105.532,105.532,105.632;115.734,115.534,115.234,115.034,114.734,114.534,114.333,114.233,114.233,114.233,114.333,114.033,113.033,111.833,110.333,108.833,108.033,107.533,107.133,106.833,106.633,106.733,106.733,106.633,106.632,106.632,106.632,106.732,106.732,106.732,106.632,106.432,106.332,106.332,106.232,106.132,105.932,105.832,105.832,105.932;115.934,115.734,115.534,115.234,114.934,114.634,114.333,114.233,114.033,113.933,113.833,113.733,113.233,111.933,110.633,109.033,108.033,107.633,107.233,106.933,106.733,106.733,106.833,106.733,106.732,106.732,106.832,106.932,107.132,107.132,107.032,106.932,106.732,106.632,106.632,106.532,106.332,106.232,106.232,106.332;116.134,115.934,115.734,115.434,115.034,114.634,114.333,114.033,113.833,113.533,113.333,113.033,112.733,111.833,110.533,109.133,108.133,107.633,107.233,107.033,106.833,106.833,106.833,106.933,106.933,106.832,106.932,107.132,107.332,107.632,107.532,107.432,107.232,107.032,107.032,107.032,106.932,106.832,106.732,106.732;116.334,116.134,115.934,115.534,115.134,114.734,114.333,113.933,113.533,113.133,112.833,112.533,112.133,111.433,110.433,109.033,108.133,107.633,107.333,107.133,106.933,106.933,107.033,107.033,107.032,107.132,107.332,107.532,107.832,107.932,107.732,107.532,107.432,107.432,107.432,107.532,107.432,107.232,107.232;116.434,116.334,116.034,115.634,115.134,114.634,114.134,113.733,113.333,112.833,112.433,112.133,111.733,111.033,109.933,108.933,108.033,107.633,107.333,107.233,107.033,107.033,107.033,107.133,107.133,107.132,107.232,107.532,107.832,108.132,108.232,108.032,107.832,107.632,107.632,107.832,107.932,108.132,107.932,107.532;116.534,116.534,116.134,115.634,115.134,114.534,114.034,113.533,113.133,112.633,112.333,112.133,111.833,110.833,109.733,108.633,108.033,107.633,107.333,107.233,107.033,107.033,107.033,107.133,107.133,107.132,107.332,107.632,107.832,108.232,108.332,108.232,108.132,107.932,107.932,108.132,108.232,108.432,108.332,108.032;117.134,116.834,116.234,115.634,115.034,114.434,113.934,113.433,112.933,112.533,112.333,112.233,112.133,110.833,109.733,108.633,107.933,107.733,107.433,107.233,107.033,106.833,106.833,106.933,106.933,107.332,107.532,107.832,108.032,108.032,108.232,108.332,108.232,108.232,108.332,108.432,108.832,108.832,108.832,108.632;116.734,116.534,116.034,115.434,114.834,114.334,113.734,113.233,112.733,112.333,112.133,112.133,110.833,109.633,108.533,107.933,107.733,107.433,107.233,107.033,106.833,106.733,106.833,107.033,107.333,107.632,107.832,108.032,108.132,108.232,108.232,108.432,108.432,108.332,108.432,108.832,109.232,109.232,109.132;116.134,115.934,115.634,115.134,114.634,114.034,113.534,113.033,112.533,112.133,111.833,111.633,111.533,110.833,109.633,108.533,107.833,107.533,107.333,107.033,106.833,106.733,106.633,106.733,106.933,107.133,107.432,107.732,107.932,108.032,108.132,108.332,108.532,108.732,108.632,108.632,108.932,109.532,109.732,109.532;115.434,11

5.334,115.034,114.734,114.234,113.834,113.334,112.834,112.333,111.833,111.433,111.133,110.933,110.333,109.433,108.333,107.733,107.433,107.133,106.933,106.733,106.633,106.633,106.733,106.733,107.033,107.332,107.632,107.932,108.032,108.232,108.432,108.532,108.732,108.932,108.832,109.032,109.632,110.132,110.032;114.634,114.534,114.434,114.134,113.934,113.634,113.234,112.734,112.233,111.633,111.133,110.833,110.533,109.933,109.133,108.133,107.433,107.233,107.033,106.633,106.533,106.633,106.633,106.733,106.733,106.833,107.132,107.432,107.732,108.032,108.232,108.532,108.632,108.732,108.932,109.032,109.132,109.532,110.132,110.632;113.834,113.834,113.734,113.734,113.634,113.534,113.234,112.734,112.133,111.533,111.033,110.633,110.233,109.633,108.833,107.833,107.133,106.933,106.833,106.533,106.333,106.433,106.533,106.633,106.733,106.833,106.932,107.332,107.632,107.932,108.232,108.532,108.632,108.732,108.832,108.932,109.332,109.732,109.932,110.432;113.134,113.134,113.234,113.334,113.534,113.634,113.334,112.734,112.133,111.533,110.933,110.433,109.933,108.933,108.133,107.533,106.933,106.633,106.433,106.333,106.133,106.233,106.333,106.533,106.633,106.733,106.933,107.232,107.532,107.932,108.232,108.532,108.732,108.732,108.832,108.932,109.232,109.832,110.132,110.232;112.634,112.634,112.734,112.934,113.234,113.534,113.434,112.634,112.033,111.533,110.833,110.133,109.433,108.433,107.633,107.033,106.633,106.333,106.233,106.033,106.033,106.133,106.133,106.333,106.533,106.633,106.833,107.132,107.532,107.832,108.132,108.532,108.732,108.732,108.932,109.032,109.332,109.932,110.332,110.432;112.134,112.334,112.334,112.434,112.634,112.334,112.234,112.334,111.734,111.333,110.633,109.733,109.033,108.333,107.533,106.833,106.333,106.033,105.933,105.933,105.933,106.033,106.033,105.933,106.233,106.633,106.833,107.032,107.432,107.732,108.132,108.432,108.632,108.632,108.832,109.232,109.532,110.132,110.532,110.632;111.734,112.034,111.934,111.834,111.934,111.634,111.534,111.634,111.634,111.133,110.633,109.733,108.733,108.033,107.333,106.633,106.233,105.933,105.733,105.733,105.833,105.933,105.933,105.933,106.033,106.433,106.733,106.932,107.232,107.632,107.932,108.232,108.532,108.532,108.732,109.132,109.532,110.132,110.732,110.932;111.634,111.734,111.634,111.234,111.234,111.434,111.334,111.334,111.334,111.233,110.733,110.033,109.033,107.833,107.133,106.533,106.033,105.833,105.733,105.833,105.833,105.833,105.933,105.733,105.933,106.333,106.633,106.833,107.132,107.532,107.832,108.032,108.232,108.432,108.632,109.132,109.532,109.932,110.532,111.132;111.434,111.434,111.434,110.934,110.834,111.034,111.134,111.134,111.034,110.933,110.633,109.933,108.933,107.833,107.033,106.333,105.933,105.733,105.633,105.733,105.733,105.733,105.833,105.733,105.833,106.233,106.433,106.733,107.032,107.432,107.732,107.932,108.032,108.232,108.532,109.032,109.432,109.832,110.232,110.832;110.834,110.934,110.934,110.734,110.534,110.734,110.834,110.734,110.533,110.233,109.733,108.733,107.433,106.633,106.333,105.933,105.733,105.633,105.433,105.533,105.433,105.533,105.633,105.733,106.133,106.433,106.633,106.932,107.332,107.632,107.832,107.932,107.932,108.232,108.732,109.232,109.732,110.132,110.432;110.434,110.434,110.434,110.234,110.034,110.234,110.134,110.034,109.934,109.533,109.133,108.533,107.133,106.333,105.933,105.633,105.533,105.433,105.333,105.333,105.233,105.333,105.433,105.633,105.933,106.333,106.533,106.732,107.132,107.432,107.632,107.832,107.832,108.032,108.532,109.032,109.532,110.032,110.332;110.034,109.934,109.834,109.734,109.634,109.434,109.434,109.234,109.134,108.934,108.633,108.333,107.733,106.833,106.033,105.733,105.433,105.133,105.133,105.033,105.033,105.033,105.133,105.233,105.433,105.733,106.133,106.333,106.632,106.932,107.232,107.532,107.632,107.732,107.932,108.332,108.832,109.532,109

.732,105.932,106.232,106.532,106.832;103.134,103.134,103.134,103.134,103.134,103.03
4,103.134,103.234,103.134,103.034,103.034,103.034,103.133,103.333,103.433,103.333,1
03.533,103.533,103.533,103.633,104.133,104.833,105.333,105.633,105.733,105.833,105.
733,105.433,105.333,105.333,105.333,105.432,105.432,105.432,105.432,105.432,105.632
,105.832,105.932,105.932;102.934,103.034,103.034,103.034,102.934,102.834,102.934,10
2.834,102.834,102.734,102.734,102.734,102.733,102.833,102.933,103.133,103.233,103.4
33,103.433,103.333,103.833,104.833,105.333,105.433,105.533,105.633,105.533,105.333,
105.233,105.233,105.233,105.232,105.232,105.232,105.232,105.232,105.432,105.532,105
.532,105.232;102.734,102.734,102.834,102.834,102.834,102.734,102.634,102.534,102.53
4,102.434,102.534,102.534,102.434,102.533,102.833,102.933,103.033,103.233,103.333,1
03.233,103.633,104.633,104.933,105.133,105.233,105.433,105.433,105.233,105.033,105.
133,104.933,104.832,105.032,105.132,105.132,105.132,105.232,105.232,105.032,104.732
;102.434,102.434,102.434,102.534,102.534,102.534,102.434,102.334,102.234,102.234,10
2.334,102.434,102.334,102.333,102.533,102.933,103.033,103.133,103.233,103.133,103.5
33,104.133,104.433,104.733,104.933,105.133,105.233,105.033,104.933,104.833,104.633,
104.233,104.432,104.832,104.832,104.832,104.932,104.632,104.432,104.232;102.134,10
2.134,102.134,102.234,102.334,102.334,102.234,102.134,102.034,102.034,102.034,102.1
34,102.234,102.133,102.333,102.733,103.033,103.033,103.133,103.133,103.433,103.733,
104.033,104.333,104.533,104.733,104.833,104.633,104.433,104.333,104.133,103.733,103
.632,104.032,104.232,104.232,104.232,103.932,103.932,103.732;102.134,102.034,101.93
4,101.834,101.934,102.034,102.034,101.934,101.834,101.834,101.834,101.934,101.934,1
01.933,102.133,102.433,102.633,102.933,103.033,103.233,103.333,103.533,103.733,104.
033,104.233,104.433,104.333,103.933,103.733,103.333,103.133,102.833,102.732,102.932
,103.132,103.132,103.132,103.132,103.132,103.532;102.134,102.034,101.734,101.634,10
1.734,101.834,101.834,101.734,101.734,101.734,101.734,101.734,101.734,101.733,101.9
33,102.233,102.433,102.733,102.933,103.033,103.233,103.433,103.633,103.833,104.033,
104.033,103.733,103.333,103.033,102.733,102.533,102.233,102.132,102.332,102.332,102
.332,102.332,102.232,102.432,102.732;102.334,102.034,101.834,101.434,101.434,101.63
4,101.634,101.734,101.634,101.634,101.634,101.634,101.634,101.534,101.733,102.133,1
02.433,102.733,102.833,102.833,103.033,103.233,103.333,103.533,103.633,103.333,103.
033,102.633,102.333,102.033,101.933,102.033,101.832,101.832,101.832,101.632,101.632
,101.632,101.732,102.232;102.434,102.134,101.734,101.334,101.234,101.334,101.434,10
1.534,101.534,101.534,101.534,101.434,101.334,101.334,101.233,101.633,102.033,102.2
33,102.433,102.433,102.533,102.733,102.933,103.033,103.033,102.733,102.533,102.133,
101.833,101.533,101.433,101.533,101.533,101.532,101.332,101.132,101.032,101.032,101
.232,101.732;102.234,102.134,101.734,101.234,101.034,101.134,101.234,101.334,101.33
4,101.334,101.334,101.234,101.134,101.034,100.933,100.933,101.233,101.533,101.633,1
01.733,101.833,101.833,102.133,102.333,102.333,102.233,102.033,101.733,101.433,101.
333,101.133,101.133,101.233,101.132,100.932,100.632,100.532,100.432,100.732,101.232
;101.934,101.834,101.634,101.134,100.834,101.034,101.034,101.134,101.034,101.034,10
1.034,101.034,100.934,100.834,100.733,100.633,100.633,100.933,101.033,101.133,101.2
33,101.333,101.333,101.533,101.633,101.533,101.533,101.233,101.133,100.933,100.733,
100.733,100.733,100.732,100.532,100.132,100.032,100.032,100.132,100.732;];

```
% Data Preprocessing: Addition of Fictitious Boundaries  
[n_rows n_cols] = size(Z);
```

```

zprime = ones(n_rows+2, n_cols+2);
zprime(2:n_rows+1, 2:n_cols+1) = Z;
zprime(1, :) = zprime(2, :); zprime(end, :) = zprime(end-1, :);
zprime(:, 1) = zprime(:, 2); zprime(:, end) = zprime(:, end-1);
Z = zprime; %we have now added a fictitious boundary all around the elevation data
%% Initial conditions of fluid depth, velocities, Rainfall and Infiltration
h_o = 0.1*ones(size(Z)); % initial flow depth (in metres)
u_o = zeros(size(Z)); % fluid initially at rest
v_o = zeros(size(Z)); % fluid initially at rest
R = zeros(size(Z)); % "rainfall" may not be the right term to use here as rain cannot just
be falling on a single pixel!
% there's no source anywhere at the start
Sourcesink = zeros(size(Z));
[N_Rows N_Cols] = size(Z);
C_o = zeros(size(Z));
% G_x = 0.0015*ones(size(Z));
% G_y = 0.0023*ones(size(Z));
G_x_o = zeros(size(Z));
G_y_o = zeros(size(Z));
%R(round(N_Rows/2+1),round(N_Cols/2)) = 2e-5; % assuming a spill source at the middle
(X mm/hr = X/(36e5) m/s)
R(10,50) = 2e-3;% a big spill before the bump
Sourcesink(round(N_Rows/2+1),round(N_Cols/2)) = 3.8e-5;

%% This section calculates the rate of Infiltration in the into the soil surface.
dldt = zeros(size(Z));
% dldt = 1e-4*ones(size(Z)); %assuming constant infiltration, infiltration rates too are in
mm/hr (very small) REF: http://www.fao.org/docrep/S8684E/s8684e0a.htm
% InfilDepth = 1e-5*ones(size(Z)); %Cumulative Infiltration Depth
% Soiltype = 'clay';
% % Choosing soil type
% if Soiltype == 'sand', soil_type = 1;
% elseif Soiltype == 'lmsy', soil_type = 2;
% elseif Soiltype == 'sdlm', soil_type = 3;
% elseif Soiltype == 'loam', soil_type = 4;
% elseif Soiltype == 'stlm', soil_type = 5;
% elseif Soiltype == 'sdcm', soil_type = 6;
% elseif Soiltype == 'cllm', soil_type = 7;
% elseif Soiltype == 'stcm', soil_type = 8;
% elseif Soiltype == 'sdcl', soil_type = 9;
% elseif Soiltype == 'stcl', soil_type = 10;
% elseif Soiltype == 'clay', soil_type = 11;
% end

%% Check that the CFL condition is satisfied, this - theoretically - ensures the stability of
the algorithm:

```

```

while del_t > max(2*grid_size./max(max(u_o + sqrt(g*h_o))),2*grid_size./max(max(v_o +
sqrt(g*h_o)))) %this is the Courant condition, may need to revise this or derive mine
    del_t = del_t/2;
end
%%
t = t + del_t;
n = 0; %start counter
while t < T
    %% The main gist of the program is here
    % The MacCormack Method is used to solve the Shallow Water equations

% [dIdt, InfilDepth] = InfiltrationModel(InfilDepth, soil_type, del_t);%this updates the
value of the infiltration rate dIdt at each time step
% if n == 0, dIdt = zeros(size(Z));end % Initial values of dIdt are unreasonable for
computational purposes.
% %h_previous = h_o; u_previous = u_o; v_previous = v_o;%stores the result of h at this
time step
% uabs = abs(u_o); vabs = abs(v_o); % Set a tolerance value for velocities at which
infiltration can occur
% % dIdt(uabs==0|vabs==0) = 0; % Infiltration doesn't take place where there's no flow
% % dIdt(h_o < 0.001) = 0; % No infiltration where the flow is less than a minimum value.
In this case, 1mm.
% dIdt(uabs<=1e-5|vabs<=1e-5) = 0; %Infiltration doesn't take place where there's slow
flow

[h_c,u_c,v_c,C_c,G_x,G_y] =
CompletoewithErosionModel(h_o,u_o,v_o,C_o,R,dIdt,del_t,grid_size,Z,t,Sourcesink,G_x_o,G_y_
o);
    h_o = h_c; u_o = u_c; v_o = v_c; C_o = C_c; G_x_o = G_x; G_y_o = G_y; %The output serve as
input at the next time step

    t = t + del_t;

    %we should turn off the source at some point just to see what happens
        if t > 60
            R(round(N_Rows/2+1),round(N_Cols/2)) = 0;
        end
    n = n + 1; %increase counter by 1
    % Check that the CFL condition is satisfied, this - theoretically - ensures the stability of
the algorithm:

    % while del_t > max(2*grid_size./max(max(u_c +
sqrt(g*h_c))),2*grid_size./max(max(v_c + sqrt(g*h_c)))) %this is the Courant condition,
may need to revise this or derive mine
    % del_t = del_t/2;
    % end

```

```

%

%% Movie of the Simulation
%Stripping the data of fictitious boundary values before displaying
udisplay = u_o(2:N_Rows-1,2:N_Cols-1);
vdisplay = v_o(2:N_Rows-1,2:N_Cols-1);
mesh(h_o(2:N_Rows-1,2:N_Cols-1))
hold on
quiver(udisplay,vdisplay)
hold off
% axis([0 N_Cols 0 N_Rows 0.10 .2]) % to prevent the plot from resizing itself
xlim([0 N_Cols]);ylim([0 N_Rows]); zlim([0 0.15]); % to prevent the plot from resizing
itself
xlabel('x'); ylabel('y');
% Check for Cancel button press
% if getappdata(h,'canceling')
% break
% end
% Report current estimate in the waitbar's message field
Percentage = (t/T)*100;
Time_so_far = t;
% waitbar(t/T,h,sprintf('%5.0f %% Elapsed time=%4.0f s',Percentage,Time_so_far))
top = title('Persistence pays');
set(top,'string',sprintf('Percentage Complete=%3.0f %%%, Elapsed time=%4.0f
s',Percentage,Time_so_far))
drawnow
pause(0.01)
end
%delete(h) % DELETE the waitbar; don't try to CLOSE it.
toc

```

E.5.1 Function files associated with the main Program in E.5

```

function [h_c,u_c,v_c,C_c,G_x,G_y] =
CompletoewithErosionModel(h_o,u_o,v_o,C_o,R,dldt,del_t,grid_size,Z,t,Sourcesink,G_x_o,G_y_
o)
%{

```

This function file is the only one needed to solve the St. Venant equations in the sense that it does not use the time-split scheme.

When split along x- and y-directions, the results are identical to those obtained here. I have merged all the four 1-D operators in the paper [Rene & Kahawita (1986)]

I adopted because of ease of debugging. The infiltration model and 2-D Erosion Model would be called at some point. They are coupled with the St. Venant equation.

This function file solves in both x and y-direction at the same time as opposed to uni-directional fractional step schemes

```

%}
%% Boundary Conditions for the Predictor Part
u_o(:,1)= u_o(:,2); v_o(1,:)= v_o(2,:); % "Open" Boundary at the Upstream
u_o(:,end)= u_o(:,end-1); v_o(end,:)= v_o(end-1,:); % "Closed" or Reflecting Boundary at the
Downstream
h_o(:,1)= h_o(:,2); h_o(1,:)= h_o(2,:); % zero-depth gradient condition i.e. specifying the
fluid-depth to be constant at the Upstream
h_o(:,end)= h_o(:,end-1); h_o(end,:)= h_o(end-1,:); % zero-depth gradient condition i.e.
specifying the fluid-depth to be constant at the Downstream
C_o(:,1)= C_o(:,2); C_o(1,:)= C_o(2,:); % No Concentration gradient across the Upstream
boundary
C_o(:,end)= C_o(:,end-1); C_o(end,:)= C_o(end-1,:); % No Concentration gradient across the
Downstream boundary

%% CONSTANTS
g = 10 ;          % acceleration due to gravity
n = 0.02;        % n is the Manning coefficient taken from Chaudhry (1993)
eddy = 0.2;      % coefficient of eddy viscosity
k = 0.1; % is a parameter used to regulate the amount of dissipation, will be determined by
trial-and-error
[N_Rows N_Cols] = size(h_o);
Manning_o = (g*n^2)./h_o.^(1/3);
Manning_o(h_o<=0) = 0; % treating zero fluid depth and/or negative depths
%del_t = del_t/2; % we take half of a time-step each time (needed when we split the
operator)
% Sourcesink = zeros(size(Z)); %assume no source or sink to the concentration profile
% Sourcesink(106,107) = 3.8e-5;
if t>10
    Sourcesink(106,107) = 0; % assume the source of pollutants was turned after t secs
end

U_o = u_o.*h_o;
V_o = v_o.*h_o;
F_o = u_o.^2.*h_o+0.5*g*h_o.^2;
G_o = u_o.*v_o.*h_o;
S_o = v_o.^2.*h_o + 0.5*g*h_o.^2;
UC_o = u_o.* C_o;
VC_o = v_o.*C_o;

Diffx = 5.93*U_o; Diffy = 5.93*V_o; % 5.93 is the value, not zero. Mixing coefficients as
suggested by Elder (1959). You can refine with Calibration
[Uoxgrad Uoygrad] = backdiffe(U_o,grid_size);
[Voxgrad Voygrad] = backdiffe(V_o,grid_size);
[Foxgrad Foygrad] = backdiffe(F_o,grid_size);
[Goxgrad Goygrad] = backdiffe(G_o,grid_size);
[S_oxgrad S_oygrad] = backdiffe(S_o,grid_size);

```

```

[Coxgrad Coygrad] = backdiffe(C_o,grid_size);
[Cu_oxgrad Cu_oygrad] = backdiffe(UC_o,grid_size);
[Cv_oxgrad Cv_oygrad] = backdiffe(VC_o,grid_size);
[Soxgrad Soygrad] = backdiffe(-Z,grid_size);
Soxgrad(Soxgrad<1e-5)= 0; Soygrad(Soygrad<1e-5)=0; %very small slopes should be
treated like a flat surface

%% The predictor step
h_p = h_o - del_t*(Uoxgrad + Voygrad - (R-dIdt));
h_p = MassConservation(h_p); % this fills in any negative depth using fluid from adjacent
cells
U_p = U_o - del_t*(Foxgrad + Goygrad)+ del_t*(g*h_o.*Soxgrad -
Manning_o.*u_o.*sqrt(u_o.^2+v_o.^2)) + eddy*del_t*backdiffe(Uoxgrad,grid_size)+
eddy*del_t*backdiffe(Uoygrad,grid_size);
V_p = V_o - del_t*(S_oygrad+ Goxgrad)+ del_t*(g*h_o.*Soygrad -
Manning_o.*v_o.*sqrt(u_o.^2+v_o.^2)) + eddy*del_t*backdiffe(Voxgrad,grid_size)+
eddy*del_t*backdiffe(Voygrad,grid_size);
C_p = C_o + del_t.*(Sourcesink + (Diffx.* backdiffe(Coxgrad,grid_size)))+(Diffy.*
backdiffe(Coygrad,grid_size))-Cu_oxgrad - Cv_oygrad);

%% Calculating Intermediate values to be used in the corrector part
u_p = U_p./h_p; % backing out u
u_p(h_p < 0.0001) = 0; % for really small fluid depths, assume the velocity is nil
v_p = V_p./h_p; % backing out v
v_p(h_p < 0.0001) = 0; % for really small fluid depths, assume the fluid is not moving at all

%% Boundary Conditions for the Corrector Part
u_p(:,1)= u_p(:,2); v_p(1,:)= v_p(2,:); % "Open" Boundary at the Upstream
u_p(:,end)= u_p(:,end-1); v_p(end,:)= v_p(end-1,:); % "Closed" or Reflecting Boundary at the
Downstream
h_p(:,1)= h_p(:,2); h_p(1,:)= h_p(2,:); % zero-depth gradient condition i.e. specifying the
fluid-depth to be constant at the Upstream
h_p(:,end)= h_p(:,end-1); h_p(end,:)= h_p(end-1,:); % zero-depth gradient condition i.e.
specifying the fluid-depth to be constant at the Downstream
C_p(:,1)= C_p(:,2); C_p(1,:)= C_p(2,:); % No Concentration gradient across the Upstream
boundary
C_p(:,end)= C_p(:,end-1); C_p(end,:)= C_p(end-1,:);% No Concentration gradient across the
Downstream boundary

F_p = u_p.^2.*h_p + 0.5*g*h_p.^2;
S_p = v_p.^2.*h_p + 0.5*g*h_p.^2;
G_p = u_p.*v_p.*h_p;
UC_p = u_o.* C_p;
VC_p = v_o.*C_p;
Diffx = 5.93*U_p; Diffy = 5.93*V_p; % 5.93 is the value, i just put zero here temporarily, as
suggested by Elder (1959). You can refine with Calibration

```

```

Manning_p = (g*n^2)./h_p.^(1/3);
Manning_p(h_p<=0) = 0; %treating zero fluid depth and/or negative depths
[Upxgrad Upygrad] = fordiffe(U_p,grid_size);
[Vpxgrad Vpygrad] = fordiffe(V_p,grid_size);
[Fpxgrad Fpygrad] = fordiffe(F_p,grid_size);
[Gpxgrad Gpygrad] = fordiffe(G_p,grid_size);
[S_pxgrad S_pygrad] = fordiffe(S_p,grid_size);
[Cpxgrad Cpygrad] = fordiffe(C_p,grid_size);
[Cu_pxgrad Cu_pygrad] = fordiffe(UC_p,grid_size);
[Cv_pxgrad Cv_pygrad] = fordiffe(VC_p,grid_size);

%% 2-D EROSION MODEL MODULE
x_Energy_Slope = Manning_p.*u_p.*sqrt(u_p.^2+v_p.^2)/g; %dividing by g because
x_Energy_Slope means g*h*Sfx whereas we only need h*Sfx here
y_Energy_Slope = Manning_p.*v_p.*sqrt(u_p.^2+v_p.^2)/g; %dividing by g because
y_Energy_Slope means g*h*Sfy whereas we only need h*Sfy here

for i = 1:N_Rows
    for j = 1:N_Cols
        x_Slope = x_Energy_Slope(i,j);
        y_Slope = y_Energy_Slope(i,j);
        [G_x(i,j), G_y(i,j)] = Two_D_erosion_model(G_x_o(i,j), G_y_o(i,j), grid_size, x_Slope,
y_Slope);

    end
end

%% The corrector step
h_c = 0.5*(h_o + h_p - del_t*(Upxgrad + Vpygrad - (R-dIdt)));
h_c = MassConservation(h_c); % this fills in any negative depth using fluid from adjacent
cells
U_c = 0.5*(U_o + U_p - del_t*(Fpxgrad + Gpygrad) + del_t*(g*h_p.*Soxgrad -
Manning_p.*u_p.*sqrt(u_p.^2+v_p.^2))+ eddy*del_t*backdiffe(Upxgrad,grid_size)+
eddy*del_t*backdiffe(Upygrad,grid_size));
V_c = 0.5*(V_o + V_p - del_t*(S_pygrad+ Gpxgrad) + del_t*(g*h_p.*Soygrad -
Manning_p.*v_p.*sqrt(u_p.^2+v_p.^2))+ eddy*del_t*backdiffe(Vpxgrad,grid_size)+
eddy*del_t*backdiffe(Vpygrad,grid_size));
C_c = 0.5*(C_o + C_p + del_t.*(Sourcesink + (Diffx.* fordiffe(Cpxgrad,grid_size)))+(Diffy.*
fordiffe(Cpygrad,grid_size))- Cu_pxgrad - Cv_pygrad));
%% Computing variables to be used in handling steep gradients (see the note in the
"artificialviscosity" file for details):
[hcx_for_grad hcy_for_grad] = fordiffe(h_c); [hcx_back_grad hcy_back_grad] =
backdiffe(h_c);
[Ucx_for_grad Ucy_for_grad] = fordiffe(U_c); [Ucx_back_grad Ucy_back_grad] =
backdiffe(U_c);

```

```

[Vcx_for_grad Vcy_for_grad] = fordiffe(V_c); [Vcx_back_grad Vcy_back_grad] =
backdiffe(V_c);
[Ccx_for_grad Ccy_for_grad] = fordiffe(C_c); [Ccx_back_grad Ccy_back_grad] =
backdiffe(C_c);
[eps_back_x_hc, eps_back_y_hc, eps_for_x_hc, eps_for_y_hc] = artificialviscosity(h_c,k);
[eps_back_x_Uc, eps_back_y_Uc, eps_for_x_Uc, eps_for_y_Uc] = artificialviscosity(U_c,k);
[eps_back_x_Vc, eps_back_y_Vc, eps_for_x_Vc, eps_for_y_Vc] = artificialviscosity(V_c,k);
[eps_back_x_Cc, eps_back_y_Cc, eps_for_x_Cc, eps_for_y_Cc] = artificialviscosity(C_c,k);
% Update the values of h, U and V:
h_c = h_c + (eps_for_x_hc.*hcx_for_grad - eps_back_x_hc.*hcx_back_grad)+
(eps_for_y_hc.*hcy_for_grad - eps_back_y_hc.*hcy_back_grad);
U_c = U_c + (eps_for_x_Uc.*Ucx_for_grad - eps_back_x_Uc.*Ucx_back_grad)+
(eps_for_y_Uc.*Ucy_for_grad - eps_back_y_Uc.*Ucy_back_grad);
V_c = V_c + (eps_for_x_Vc.*Vcx_for_grad - eps_back_x_Vc.*Vcx_back_grad)+
(eps_for_y_Vc.*Vcy_for_grad - eps_back_y_Vc.*Vcy_back_grad);
C_c = C_c + (eps_for_x_Cc.*Ccx_for_grad - eps_back_x_Cc.*Ccx_back_grad)+
(eps_for_y_Cc.*Ccy_for_grad - eps_back_y_Cc.*Ccy_back_grad);
u_c = U_c./h_c; %backing out u
u_c(h_c < 0.0001) = 0; %we assume fluid isn't moving when h is small e.g. less than 1 mm
v_c = V_c./h_c; %backing out v
v_c(h_c < 0.0001) = 0; %we assume fluid isn't moving when h is small e.g. less than 1 mm
%% 2-D EROSION MODEL MODULE
Manning_c = (g*n^2)./h_c.^(1/3);
Manning_c(h_c<=0) = 0; %treating zero fluid depth and/or negative depths
x_Energy_Slope = Manning_c.*u_c.*sqrt(u_c.^2+v_c.^2)/g; % dividing by g because
x_Energy_Slope means g*h*Sfx whereas we only need h*Sfx here
y_Energy_Slope = Manning_c.*v_c.*sqrt(u_c.^2+v_c.^2)/g; % dividing by g because
y_Energy_Slope means g*h*Sfy whereas we only need h*Sfy here

for i = 1:N_Rows
    for j = 1:N_Cols
        x_Slope = x_Energy_Slope(i,j);
        y_Slope = y_Energy_Slope(i,j);
        [G_x(i,j), G_y(i,j)] = Two_D_erosion_model(G_x_o(i,j), G_y_o(i,j), grid_size, x_Slope,
y_Slope);

    end
end

```

```

function [dIdt, InfilDepth] = InfiltrationModel(InfilDepth, soil_type, del_t)
%{
This function file helps to compute the Rate of Infiltration, dIdt, and
the cumulative Infiltrative Depth, InfilDepth. Its input is Initial
Infiltration Depth (at the start of the time step), the soil type and time
step. The parameters for a given soil type has been hardwired into the

```


code. Soil data was obtained from Rawls, W.J., and D.L.Brakensiek. 1983. A procedure to predict Green and Ampt. infiltration parameters. In Advances in infiltration. Proc. of the Nat'l Conference on Advances in Infiltration. Dec. 12-13. Parameters for each soil type are hardwired and can be changed based on observation or sound science e.g. Sensitivity Analysis. The simple Infiltration Model can be found in Groves (1989): Groves, J. R. (1989), A Practical Soil Moisture Profile Model, Journal of the American Water Resources Association, 25, 4, 875-880. The model is an ordinary differential equation so ode45 was used in solving it.

```
%}
```

```
%Antecedent Moisture can be any value, 0.3 is used as an example
AntecedentMoisture = 0.3*ones(size(InfilDepth));
Lambda = 0.23;
```

```
%% Soil Parameters
if soil_type == 1
    SatMoisture = 0.437*ones(size(InfilDepth));
    ResidMoisture = 0.02*ones(size(InfilDepth));
    AirEntry = 0.0495;
    HydCond = 3.2984e-5;
elseif soil_type == 2
    SatMoisture = 0.437*ones(size(InfilDepth));
    ResidMoisture = 0.036*ones(size(InfilDepth));
    AirEntry = 0.0613;
    HydCond = 8.372e-6;
elseif soil_type == 3
    SatMoisture = 0.453*ones(size(InfilDepth));
    ResidMoisture = 0.041*ones(size(InfilDepth));
    AirEntry = 0.1101;
    HydCond = 3.052e-6;
elseif soil_type == 4
    SatMoisture = 0.463*ones(size(InfilDepth));
    ResidMoisture = 0.029*ones(size(InfilDepth));
    AirEntry = 0.0889;
    HydCond = 9.52e-7;
elseif soil_type == 5
    SatMoisture = 0.501*ones(size(InfilDepth));
    ResidMoisture = 0.015*ones(size(InfilDepth));
    AirEntry = 0.1668;
    HydCond = 1.82e-6;
elseif soil_type == 6
    SatMoisture = 0.398*ones(size(InfilDepth));
    ResidMoisture = 0.068*ones(size(InfilDepth));
    AirEntry = 0.2185;
    HydCond = 4.2e-7;
```

```

elseif soil_type == 7
    SatMoisture = 0.464*ones(size(InfilDepth));
    ResidMoisture = 0.155*ones(size(InfilDepth));
    AirEntry = 0.2088;
    HydCond = 2.8e-7;
elseif soil_type == 8
    SatMoisture = 0.471*ones(size(InfilDepth));
    ResidMoisture = 0.039*ones(size(InfilDepth));
    AirEntry = 0.273;
    HydCond = 2.8e-7;
elseif soil_type == 9
    SatMoisture = 0.43*ones(size(InfilDepth));
    ResidMoisture = 0.109*ones(size(InfilDepth));
    AirEntry = 0.239;
    HydCond = 1.68e-7;
elseif soil_type == 10
    SatMoisture = 0.479*ones(size(InfilDepth));
    ResidMoisture = 0.056*ones(size(InfilDepth));
    AirEntry = 0.2922;
    HydCond = 1.4e-7;
elseif soil_type == 11
    SatMoisture = 0.475*ones(size(InfilDepth));
    ResidMoisture = 0.09*ones(size(InfilDepth));
    AirEntry = 0.3163;
    HydCond = 8.4e-8;
end

%% The model is applied below.

%Compute the Wetting Front suction head, wf
wf=(AirEntry./((AntecedentMoisture - ResidMoisture)./...
    (SatMoisture - ResidMoisture)).^(1/Lambda));

% Convert the arrays into vectors so that they can be passed into ode45

InfilDepth = InfilDepth(:); %this turns a matrix into a vector
SatMoisture = SatMoisture(:);
AntecedentMoisture = AntecedentMoisture(:);
wf = wf(:);

% Declare the function handle for the ODE solver
Didt = @(t_inf, InfilDepth) HydCond./(1-exp(-InfilDepth./...
    ((SatMoisture - AntecedentMoisture).*wf));
[~, InfilDepth] = ode45(Didt,[0 del_t],InfilDepth(:));

% Only the last row of InfilDepth is needed

```

```

InfilDepth = InfilDepth(end, :);

% Convert the vectors back into arrays

InfilDepth = reshape(InfilDepth,size(ResidMoisture));
SatMoisture = reshape(SatMoisture,size(ResidMoisture));
AntecedentMoisture = reshape(AntecedentMoisture,size(ResidMoisture));
wf = reshape(wf, size(ResidMoisture));

% Compute the Rate of Infiltration using the simple model in Groves (1989)
dIdt = HydCond./(1-exp(-InfilDepth./...
    ((SatMoisture - AntecedentMoisture).*wf)));

function [G_x, G_y] = Two_D_erosion_model(G_x_o, G_y_o, grid_size, x_Slope, y_Slope)

% Developing the code for the 2-D Erosion Model

% clc
% close all
% clear all

% These are parameters that can be calibrated when data is available

kt = 0.75; % kt is a transport coefficient [L^0.5 T^2 M ^-0.5]. Its value is usually between
0.5 and 1.0
kr = 0.0115; % rill erodibility parameter (s/m). Default value in WEPP is 0.0115
gamma = 9807; % specific weight of water (= density * g), N/m^3. i.e. [ML^-2T^-2]
Tau_c = 3.1; % critical shear stress (N/m^2)

% The following are commented out because they were only used in developing
% the script file
%h = 0.001; % fluid depth
%grid_size = 5; % square grids are used so del_x = del_y = grid size
%S_f_x = 0.002; % friction slope in x-direction
%S_f_y = 0.004; % friction slope in y-direction
%G_x = 0.0023;
%G_y = 0.0015;

alpha = 8;
%{
alpha is a first order reaction coefficient for deposition.
alpha = Vf/q where Vf is the particle fall velocity and q is the unit with
discharge. Its unit is m^-1. I am choosing a value of 8 because it is the
average value (3-13) observed by Foster, G.R. [Ref: Journal Title: Hydrologic
modeling of small watersheds. Chapter 8. Modeling the erosion process. pp.
297-360. 1982]. If I use the formula Vf/q, I arrive at unrealistically high

```

values for alpha because q is typically low. So I am making alpha a parameter to be calibrated to fit whatever problem at hand
 %}

%{

If you decide not to make alpha a parameter but to calculate it, you can uncomment this section.

u = 0.02; % x-direction velocity

v = 0.01; % y-direction velocity

sgs = 2.65; % specific gravity of sediment

nu = 1e-6; % Kinematic viscosity

g = 9.81; % acceleration due to the gravity

d50 = 0.000707; % Median particle diameter, m

% Calculate Fall velocity using Rubey's equation

R1 = 36*nu^2/(g*d50^3*(sgs-1));

R2 = sqrt(2/3 + R1) - sqrt(R1);

Vf = R2*sqrt((sgs-1)*g*d50); % Fall velocity

qx = u*h;

qy = v*h;

alpha_x = Vf/qx; alpha_y = Vf/qy;

%}

% x_Slope is the product of the fluid depth and energy slope i.e. h*Sfx

% y_Slope is h*Sfy

% The flow shear stress is the product of the specific weight of water, the

% flow depth and energy slope i.e. Tau_f = gamma*h*Sf

Tau_f_x = gamma*abs(x_Slope); % flow shear stress acting on soil particles in the x-direction (N/m^2)

Tau_f_y = gamma*abs(y_Slope); % flow shear stress acting on soil particles in the y-direction (N/m^2)

T_c_x = kt*Tau_f_x^(3/2);

T_c_y = kt*Tau_f_y^(3/2);

D_c_x = kr*(Tau_f_x - Tau_c); % a variable that may be used later

D_c_y = kr*(Tau_f_y - Tau_c); % a variable that may be used later

% Solve for the x-direction

if abs(G_x_o) > T_c_x

 % compute net deposition

 D_f_x = alpha*(T_c_x - abs(G_x_o)); % this value should be negative

else

 % compute net detachment

 if Tau_f_x > Tau_c

 D_f_x = D_c_x*(1 - abs(G_x_o)/T_c_x);

 else

```

        D_f_x = 0;
    end
end

% Update the value of the sediment load in the x-direction
G_x = G_x_o + D_f_x*grid_size;

% Solve for the y-direction
if abs(G_y_o) > T_c_y
    % compute net deposition
    D_f_y = alpha*(T_c_y - abs(G_y_o)); % this value should be negative
else
    % compute net detachment
    if Tau_f_y > Tau_c
        D_f_y = D_c_y*(1 - abs(G_y_o)/T_c_y);
    else
        D_f_y = 0;
    end
end
end

% Update the value of the sediment load in the y-direction
G_y = G_y_o + D_f_y*grid_size;

```

```

function h_o = MassConservation(h_o)
%% Mass Conservation Subroutine.
%{
This section seeks to conserve mass flow in the system by eliminating the
appearance of negative depths.
Details of the scheme can be found in the reference paper: OVERLAND FLOW
AND INFILTRATION MODELLING FOR SMALL PLOTS DURING UNSTEADY
RAIN:NUMERICAL RESULTS VERSUS OBSERVED VALUES by ESTEVES, M et al, Journal of
Hydrology Vol. 228, 265-282 (2000)
The paper recalculates for the negative flow depths h_o by filling them in from the
adjacent cells with non- negative depth.
%}

h_negative = h_o; %Set Initial value of testing matrix h_neg from input matrix h_o provided.
[N_Rows N_Cols] = size(h_o);
% This routine is only valid for interior points of the matrix. Results not
% valid for matrix edges.
for i = 2:N_Rows-1; % This loop accounts for operations carried out on the interior rows of
the matix.
    for j = 2:N_Cols-1; % This loop operates on the interior columns of the matrix.
        if h_negative(i,j) < 0; % Test for negative depth in cell

```

```

    h_sum = h_negative(i-1, j)+ h_negative(i+1, j) + h_negative(i, j-1)+ h_negative(i,
j+1);% Estimates the sum of the four adjacent cells to the test cell.
    h_deficit = abs(h_negative(i,j))- h_sum; % Estimate Deficit/Surplus value Sum of
Adjacent nodes is sufficient to fill the negative depth encountered.
    if h_deficit <= 0; % h_sum sufficient to fill depth
        h_negative(i-1, j)= h_negative(i-1, j)- abs(h_negative(i,j))*h_negative(i-1, j)/h_sum;
% Update for cell depth based on cell's fractional contribution to the sum(h_sum)
        h_negative(i+1, j) = h_negative(i+1, j)- abs(h_negative(i,j))*h_negative(i+1,
j)/h_sum; % Update for cell depth based on cell's fractional contribution to the sum(h_sum)
        h_negative(i, j-1) = h_negative(i, j-1)- abs(h_negative(i,j))*h_negative(i, j-1)/h_sum;
% Update for cell depth based on cell's fractional contribution to the sum(h_sum)
        h_negative(i, j+1)= h_negative(i, j+1)- abs(h_negative(i,j))*h_negative(i,
j+1)/h_sum; % Update for cell depth based on cell's fractional contribution to the
sum(h_sum)
        h_negative(i,j) = 0; % Fill up cell from adjacent cell contributions.
    else % if h_def > 0 i.e Positive Deficit still exists after withdrawal of h_sum
        h_sum_diagonal = h_negative(i-1, j-1)+ h_negative(i-1, j+1)+ h_negative(i+1, j-
1)+h_negative(i+1, j+1);% Calculates the sum of the 4 diagonal cells to be used to fill in
deficit depth (h_def)
        h_negative(i-1,j) = 0; h_negative(i+1,j) =0; h_negative(i,j-1)=0; h_negative(i,j+1)=
0; % Zero values in adjacent cells showing that total withdrawal has been achieved.
        h_negative(i-1, j-1)= h_negative(i-1, j-1)- h_deficit *h_negative(i-1, j-
1)/h_sum_diagonal; % Update for diagonal cell depth based on cell's fractional contribution
to the sum(h_sum_diagonal)
        h_negative(i+1, j-1) = h_negative(i+1, j-1)- h_deficit *h_negative(i+1, j-
1)/h_sum_diagonal;% Update for diagonal cell depth based on cell's fractional contribution
to the sum(h_sum_diagonal)
        h_negative(i-1, j+1) = h_negative(i-1, j+1)- h_deficit *h_negative(i-1,
j+1)/h_sum_diagonal;% Update for diagonal cell depth based on cell's fractional
contribution to the sum(h_sum_diagonal)
        h_negative(i+1, j+1)= h_negative(i+1, j+1)- h_deficit *h_negative(i+1,
j+1)/h_sum_diagonal;% Update for diagonal cell depth based on cell's fractional
contribution to the sum(h_sum_diagonal)
        h_negative(i,j) = 0; % Fill up cell from diagonal cell contributions
    end
end
end
end
h_o = h_negative; % Return the value of h_o from the recalculated h_neg matrix.

```

```

function [Ax, Ay] = backdiffe(A,grid_size)

```

```

%{

```

```

We are employing the finite difference formula found in Hoffman (1992)
"Numerical Methods for Engineers and Scientists" pg. 177 in place of using
the "gradient" function or the earlier "shifting" method we adopted.

```

```

%}
if nargin < 2, grid_size = 1; end
spacing = grid_size; %we assume square grids so del_x=del_y and the formula
    %uses del_x and del_y as the spacing
Ax = zeros(size(A));
Ay = Ax;
% Backward Difference
Ay(2:end,:) = A(2:end,:)-A(1:end-1,:);
%the backward difference works up to the (end-1)th column so we employ the
%forward difference to help us finish the job
Ay(1,:)=A(2,:)-A(1,:);

Ax(:,2:end)= A(:,2:end)-A(:,1:end-1);
%the backward difference works up to the (end-1)th row so we employ the
%forward difference to help us finish the job
Ax(:,1) = A(:,2)-A(:,1);
Ax = Ax/spacing;
Ay = Ay/spacing;

```

```

function [Ax, Ay] = fordiffe(A,grid_size)
%{
We are employing the finite difference formula found in Hoffman (1992)
"Numerical Methods for Engineers and Scientists" pg. 177 in place of using
the "gradient" function or the earlier "shifting" method we adopted.
%}
if nargin < 2, grid_size = 1; end
spacing = grid_size; %we assume square grids so del_x=del_y and the formula
    %uses del_x and del_y as the spacing
Ax = zeros(size(A));
Ay = Ax;
% Forward Difference
Ay(1:end-1,:)= A(2:end,:)-A(1:end-1,:);
%the forward difference works up to the (end-1)th row so we employ the
%backward difference to help us finish the job
Ay(end,:) = A(end,:)-A(end-1,:);
Ax(:,1:end-1) = A(:,2:end)-A(:,1:end-1);
%the forward difference works up to the (end-1)th column so we employ the
%backward difference to help us finish the job
Ax(:,end)= A(:,end)- A(:,end-1);

Ax = Ax/spacing;
Ay = Ay/spacing;

```



```

%Sourcesink = zeros(size(Z));
[NRows NCols] = size(Z);
%C_o = zeros(size(Z));
% G_x = 0.0015*ones(size(Z));
% G_y = 0.0023*ones(size(Z));
%G_x_o = zeros(size(Z));
%G_y_o = zeros(size(Z));
%R(round(NRows/2+1),round(NCols/2)) = 2e-3; % assuming a spill source at the middle
(X mm/hr = X/(36e5) m/s)
R(10,50) = 2e-2;% a big spill before the bump
%Sourcesink(round(N_Rows/2+1),round(N_Cols/2)) = 3.8e-5;

%% This section calculates the rate of Infiltration in the into the soil surface.
I = zeros(size(Z)); %I is rate!!
% dIdt = 1e-4*ones(size(Z)); %assuming constant infiltration, infiltration rates too are in
mm/hr (very small) REF: http://www.fao.org/docrep/S8684E/s8684e0a.htm
% InfilDepth = 1e-5*ones(size(Z)); %Cumulative Infiltration Depth
% Soiltype = 'clay';
% % Choosing soil type
% if Soiltype == 'sand',    soil_type = 1;
% elseif Soiltype == 'lmsy',    soil_type = 2;
% elseif Soiltype == 'sdlm',    soil_type = 3;
% elseif Soiltype == 'loam',    soil_type = 4;
% elseif Soiltype == 'stlm',    soil_type = 5;
% elseif Soiltype == 'sdcml',    soil_type = 6;
% elseif Soiltype == 'cllm',    soil_type = 7;
% elseif Soiltype == 'stcm',    soil_type = 8;
% elseif Soiltype == 'sdcl',    soil_type = 9;
% elseif Soiltype == 'stcl',    soil_type = 10;
% elseif Soiltype == 'clay',    soil_type = 11;
% end

%% Check that the CFL condition is satisfied, this - theoretically - ensures the stability of
the algorithm:
while del_t > max(2*grid_size./max(max(u_o + sqrt(g*h_o))),2*grid_size./max(max(v_o +
sqrt(g*h_o)))) %this is the Courant condition, may need to revise this or derive mine
    del_t = del_t/2;
end
%%
t = t + del_t;
n = 0; %start counter
while t < T
    %% The main gist of the program is here
    % The MacCormack Method is used to solve the Shallow Water equations

```

```

% [dIdt, InfilDepth] = InfiltrationModel(InfilDepth, soil_type, del_t);%this updates the
value of the infiltration rate dIdt at each time step
% if n == 0, dIdt = zeros(size(Z));end % Initial values of dIdt are unreasonable for
computational purposes.
% %h_previous = h_o; u_previous = u_o; v_previous = v_o;%stores the result of h at this
time step
% uabs = abs(u_o); vabs = abs(v_o); % Set a tolerance value for velocities at which
infiltration can occur
% % dIdt(uabs==0|vabs==0) = 0; % Infiltration doesn't take place where there's no
flow
% % dIdt(h_o < 0.001) = 0; % No infiltration where the flow is less than a minimum
value. In this case, 1mm.
% dIdt(uabs<=1e-5|vabs<=1e-5) = 0; %Infiltration doesn't take place where there's
slow flow

[h_c,u_c,v_c] = first_x_operator(h_o,u_o,v_o,del_t,grid_size,Z,R,I,NRows,NCols);
h_o = h_c; u_o = u_c; v_o = v_c; %The output serve as input at the next time step
[h_c,u_c,v_c] = first_y_operator(h_o,u_o,v_o,del_t,grid_size,Z,R,I,NRows,NCols);
h_o = h_c; u_o = u_c; v_o = v_c; %The output serve as input at the next time step

t = t + del_t;

%we should turn off the source at some point just to see what happens
% if t> 600
% R(round(NRows/2+1),round(NCols/2)) = 0;
% end
n= n+ 1; %increase counter by 1
% Check that the CFL condition is satisfied, this - theoretically - ensures the stability of
the algorithm:

% while del_t > max(2*grid_size./max(max(u_c +
sqrt(g*h_c))),2*grid_size./max(max(v_c + sqrt(g*h_c)))) %this is the Courant condition,
may need to revise this or derive mine
% del_t = del_t/2;
% end
%

%% Movie of the Simulation
%Stripping the data of fictitious boundary values before displaying
udisplay = u_o(2:NRows-1,2:NCols-1);
vdisplay = v_o(2:NRows-1,2:NCols-1);
mesh(h_o(2:NRows-1,2:NCols-1))
hold on
quiver(udisplay,vdisplay)
hold off
% axis([0 N_Cols 0 N_Rows 0.10 .2]) % to prevent the plot from resizing itself

```

```

    xlim([0 NCols]);ylim([0 NRows]); zlim([0.1 0.11]); % to prevent the plot from resizing
    itself
    xlabel('x'); ylabel('y');
    % Check for Cancel button press
    % if getappdata(h,'canceling')
    %     break
    % end
    % Report current estimate in the waitbar's message field
    Percentage = (t/T)*100;
    Time_so_far = t;
    % waitbar(t/T,h,sprintf('%5.0f %% Elapsed time=%4.0f s',Percentage,Time_so_far))
    top = title('Persistence pays');
    set(top,'string',sprintf('Percentage Complete=%3.0f %%%, Elapsed time=%4.0f
s',Percentage,Time_so_far))
    drawnow
    pause(0.01)
end
%delete(h) % DELETE the waitbar; don't try to CLOSE it.
toc

```

E.6.1 Function files associated with the main Program in E.6

```

function [h_c,u_c,v_c] = first_x_operator(h_o,u_o,v_o,del_t,grid_size,Z,R,I,NRows,NCols)
%{
This function file is the First Lx operator. It advances the solution in
the x-direction by half a time-step. Its output will serve as inputs to the
First Ly operator(after Garcia & Kahawita, 1986.
%}

%% Boundary Conditions for the Predictor Part
% The boundary is either closed or open. Only one can be used at a time.
% This means we will have the closed boundary conditions commented out when
% we need the program to model an open boundary and vice versa.
g = 10 ; % acceleration due to gravity
n = 0.03 ; % n is Manning coefficient
%del_t = del_t/2; % we take half of a time-step each time

F_o = u_o.^2.*h_o + 0.5*g*h_o.^2;
G_o = u_o.*v_o.*h_o;
Sfx_o = n^2*u_o.*sqrt(u_o.^2 + v_o.^2)./h_o.^(4/3);
U_o = u_o.*h_o; % The Unit width discharge in the x-direction
V_o = v_o.*h_o; % The Unit width discharge in the y-direction

% This is the reflection (CLOSED) boundary condition where (i-1,j) is replaced by (i+1,j)
and

```

```

% the sign of the normal velocity component is swiched.

% % I am wary of pre-allocating here because the values I used in
% % pre-allocating will inadvertently be used in the computation causing
% % errors. So I will trade speed (and memory) for accuracy.
% for i = 1
%   for j = 1:NCols
%     h_p(i,j) = h_o(i,j) - (del_t/grid_size).*(U_o(i,j) + U_o(i+1,j)) + del_t.*(R(i,j)-I(i,j));
%     U_p(i,j) = U_o(i,j) - (del_t/grid_size).*(F_o(i,j) - F_o(i+1,j)) + del_t*g*0.5*(h_o(i,j) +
h_o(i+1,j)).*(-(Z(i,j) - Z(i+1,j))/grid_size - Sfx_o(i,j));
%     V_p(i,j) = V_o(i,j) - (del_t/grid_size).*(G_o(i,j) + G_o(i+1,j));
%   end
% end

% This is the transmitting (OPEN) boundary condition where (i-1,j) is
% replaced by (i+1,j) and the sign of the normal component of velocity is
% unchanged

for i = 1
  for j = 1:NCols
    h_p(i,j) = h_o(i,j) - (del_t/grid_size).*(U_o(i,j) - U_o(i+1,j)) + del_t.*(R(i,j)-I(i,j));
    U_p(i,j) = U_o(i,j) - (del_t/grid_size).*(F_o(i,j) - F_o(i+1,j)) + del_t*g*0.5*(h_o(i,j) +
h_o(i+1,j)).*(-(Z(i,j) - Z(i+1,j))/grid_size - Sfx_o(i,j));
    V_p(i,j) = V_o(i,j) - (del_t/grid_size).*(G_o(i,j) - G_o(i+1,j));
  end
end
% Pre-allocate with boundary values.
%Pre-allocating here fixes the size of the matrices and ensures they don't
%change once they get inside the for loops below. If I don't pre-allocate
%here, the first row would be replaced by zeros since I am starting the
%computation at i = 2, so MATLAB simply fills i = 1 with zeros which I
%don't want.I am using the first row to pre-allocate since that is the only known
"Predicted" value so far

h_p = repmat(h_p(1,:),NRows,1); % fill up with matrix with this boundary value
U_p = repmat(U_p(1,:),NRows,1); % It is a nice way to pre-allocate also
V_p = repmat(V_p(1,:),NRows,1);

%%% CONSTANTS
k = 0.1; % is a parameter used to regulate the amount of dissipation, will be determined by
trial-and-error

%%% The predictor step (backward difference)
% Calculate the interior parts

for i = 2:NRows

```

```

for j = 1:NCols
    h_p(i,j) = h_o(i,j) - (del_t/grid_size).*(U_o(i,j)-U_o(i-1,j))+ del_t.*(R(i,j)-I(i,j));
    U_p(i,j) = U_o(i,j) - (del_t/grid_size).*(F_o(i,j)-F_o(i-1,j))+ del_t*g*0.5*(h_o(i,j)+h_o(i-1,j)).*(-(Z(i,j)-Z(i-1,j))/grid_size - Sfx_o(i,j));
    V_p(i,j) = V_o(i,j) - (del_t/grid_size).*(G_o(i,j)-G_o(i-1,j));
end
end

%% Calculate Intermediate Values to be used in the Corrector Part
u_p = U_p./h_p;    % backing out u
v_p = V_p./h_p;    % backing out v

F_p = u_p.^2.*h_p + 0.5*g*h_p.^2;
G_p = u_p.*v_p.*h_p;
Sfx_p = n^2*u_p.*sqrt(u_p.^2 + v_p.^2)./h_p.^(4/3);

%% The corrector step (forward difference)

% % Calculate the Boundary values
% % For the CLOSED Boundary condition. Always notice the sign changes in U and
% %G but not F
% for i = NRows
%   for j = 1:NCols
%     h_c(i,j) = 0.5 * (h_o(i,j) + h_p(i,j) - (del_t/grid_size).*(-U_p(i-1,j)-U_p(i,j)) +
del_t.*(R(i,j)-I(i,j)));
%     U_c(i,j) = 0.5 * (U_o(i,j) + U_p(i,j) - (del_t/grid_size).*(F_p(i-1,j)-F_p(i,j)) +
del_t*g*0.5*(h_p(i-1,j) + h_p(i,j)).*(-(Z(i-1,j) - Z(i,j))/grid_size - Sfx_p(i,j)));
%     V_c(i,j) = 0.5 * (V_o(i,j) + V_p(i,j) - (del_t/grid_size).*(-G_p(i-1,j)-G_p(i,j)));
%   end
% end

% For the OPEN Boundary condition, no sign changes.
for i = NRows
    for j = 1:NCols
        h_c(i,j) = 0.5 * (h_o(i,j) + h_p(i,j) - (del_t/grid_size).*(U_p(i-1,j)-U_p(i,j)) + del_t.*(R(i,j)-I(i,j)));
        U_c(i,j) = 0.5 * (U_o(i,j) + U_p(i,j) - (del_t/grid_size).*(F_p(i-1,j)-F_p(i,j)) + del_t*g*0.5*(h_p(i-1,j) + h_p(i,j)).*(-(Z(i-1,j) - Z(i,j))/grid_size - Sfx_p(i,j)));
        V_c(i,j) = 0.5 * (V_o(i,j) + V_p(i,j) - (del_t/grid_size).*(G_p(i-1,j)-G_p(i,j)));
    end
end
end
% Pre-allocate with boundary values.
%Pre-allocating here fixes the size of the matrices and ensures they don't
%change once they get inside the for loops below. If I don't pre-allocate
%here, the resulting matrices from the for loops below would have a size of
%[NRows - 1, NCols] instead of the normal [NRows, NCols]. I am using the

```

```
%last row (i = NRows)to pre-allocate since that is the only known "corrected" value so
%far.
```

```
h_c = repmat(h_c(NRows,:),NRows,1); % fill up with matrix with this boundary value
```

```
U_c = repmat(U_c(NRows,:),NRows,1);
```

```
V_c = repmat(V_c(NRows,:),NRows,1);
```

```
for i = 1:NRows-1
```

```
    for j = 1:NCols
```

```
        h_c(i,j) = 0.5 * (h_o(i,j) + h_p(i,j) - (del_t/grid_size).*(U_p(i+1,j)-U_p(i,j)) + del_t.*(R(i,j)-
I(i,j)));
```

```
        U_c(i,j) = 0.5 * (U_o(i,j) + U_p(i,j) - (del_t/grid_size).*(F_p(i+1,j)-F_p(i,j)) +
del_t*g*0.5*(h_p(i+1,j) + h_p(i,j)).*(-(Z(i+1,j)-Z(i,j))/grid_size - Sfx_p(i,j)));
```

```
        V_c(i,j) = 0.5 * (V_o(i,j) + V_p(i,j) - (del_t/grid_size).*(G_p(i+1,j)-G_p(i,j)));
```

```
    end
```

```
end
```

```
u_c = U_c./h_c;    % backing out u
```

```
v_c = V_c./h_c;    % backing out v
```

```
%% Computing variables to be used in handling steep gradients (see the note in the
"artificialviscosity" file for details):
```

```
[eps_back_x_hc, eps_for_x_hc] = artificialviscosity2D_x(h_c,k);
```

```
% [eps_back_x_uc, eps_for_x_uc] = artificialviscosity2D(u_c,k);
```

```
% [eps_back_x_vc, eps_for_x_vc] = artificialviscosity2D(v_c,k);
```

```
% Update the INTERIOR values of h,u and v:
```

```
for i = 2:NRows-1
```

```
    for j = 1:NCols
```

```
        h_c(i,j) = h_c(i,j) + eps_for_x_hc(i,j).*(h_c(i+1,j) - h_c(i,j)) - eps_back_x_hc(i,j).*(h_c(i,j) - h_c(i-
1,j));
```

```
        % u_c(i,j) = u_c(i,j) + eps_for_x_uc(i,j).*(u_c(i+1,j) - u_c(i,j)) - eps_back_x_uc(i,j).*(u_c(i,j) -
u_c(i-1,j));
```

```
        % v_c(i,j) = v_c(i,j) + eps_for_x_vc(i,j).*(v_c(i+1,j) - v_c(i,j)) - eps_back_x_vc(i,j).*(v_c(i,j) -
v_c(i-1,j));
```

```
    end
```

```
end
```

```
end
```

```
function [h_c,u_c,v_c] = first_y_operator(h_o,u_o,v_o,del_t,grid_size,Z,R,I,NRows,NCols)
```

```
%{
```

```
This function file is the First Ly operator. It advances the solution in
```

the y-direction by half a time-step. Its output will serve as inputs to the Second Ly operator (after Garcia & Kahawita, 1986).

```

%}

%% Boundary Conditions for the Predictor Part
% The boundary is either closed or open. Only one can be used at a time.
% This means we will have the closed boundary conditions commented out when
% we need the program to model an open boundary and vice versa.
g = 10 ;           % acceleration due to gravity
n = 0.03 ;        % n is Manning coefficient
%del_t = del_t/2; % we take half of a time-step each time

S_o = v_o.^2.*h_o + 0.5*g*h_o.^2;
G_o = u_o.*v_o.*h_o;
Sfy_o = n^2*v_o.*sqrt(u_o.^2 + v_o.^2)./h_o.^(4/3);
U_o = u_o.*h_o; % The Unit width discharge in the x-direction
V_o = v_o.*h_o; % The Unit width discharge in the y-direction

% % This is the reflection (closed) boundary condition where (i,j-1) is replaced by (i,j+1)
% and
% % the sign of the normal velocity component is switched.
%
% % I am wary of pre-allocating here because the values I used in
% % pre-allocating will inadvertently be used in the computation causing
% % errors. So I will trade speed (and memory) for accuracy.
% for i = 1:NRows
%   for j = 1
%     h_p(i,j) = h_o(i,j) - (del_t/grid_size).*(V_o(i,j) + V_o(i,j+1)) + del_t.*(R(i,j)-I(i,j));
%     U_p(i,j) = U_o(i,j) - (del_t/grid_size).*(G_o(i,j) + G_o(i,j+1));
%     V_p(i,j) = V_o(i,j) - (del_t/grid_size).*(S_o(i,j) - S_o(i,j+1)) + del_t*g*0.5*(h_o(i,j) +
% h_o(i,j+1)).*(-(Z(i,j) - Z(i,j+1))/grid_size - Sfy_o(i,j));
%   end
% end

% This is the transmitting (OPEN) boundary condition where (i,j-1) is
% replaced by (i,j+1) and the sign of the normal component of velocity is
% unchanged

for i = 1:NRows
  for j = 1
    h_p(i,j) = h_o(i,j) - (del_t/grid_size).*(V_o(i,j) - V_o(i,j+1)) + del_t.*(R(i,j)-I(i,j));
    U_p(i,j) = U_o(i,j) - (del_t/grid_size).*(G_o(i,j) - G_o(i,j+1));
    V_p(i,j) = V_o(i,j) - (del_t/grid_size).*(S_o(i,j) - S_o(i,j+1)) + del_t*g*0.5*(h_o(i,j) +
h_o(i,j+1)).*(-(Z(i,j) - Z(i,j+1))/grid_size - Sfy_o(i,j));
  end
end

```

```

% Pre-allocate with boundary values.
%Pre-allocating here fixes the size of the matrices and ensures they don't
%change once they get inside the for loops below. If I don't pre-allocate
%here, the first column would be replaced by zeros since I am starting the
%computation at j = 2, so MATLAB simply fills j = 1 with zeros which I
%don't want.I am using the first row to pre-allocate since that is the only known
"Predicted" value so far

h_p = repmat(h_p(:,1),1,NCols); % fill up with matrix with this boundary value
U_p = repmat(U_p(:,1),1,NCols); % It is a nice way to pre-allocate also
V_p = repmat(V_p(:,1),1,NCols);

%%% CONSTANTS
k = 0.1; % is a parameter used to regulate the amount of dissipation, will be determined by
trial-and-error

%%% The predictor step (backward difference)
% Calculate the interior parts

for i = 1:NRows
    for j = 2:NCols
        h_p(i,j) = h_o(i,j) - (del_t/grid_size).*(V_o(i,j)-V_o(i,j-1))+ del_t.*(R(i,j)-I(i,j));
        U_p(i,j) = U_o(i,j) - (del_t/grid_size).*(G_o(i,j)-G_o(i,j-1));
        V_p(i,j) = V_o(i,j) - (del_t/grid_size).*(S_o(i,j)-S_o(i,j-1))+ del_t*g*0.5*(h_o(i,j)+h_o(i,j-
1)).*(-(Z(i,j)-Z(i,j-1))/grid_size - Sfy_o(i,j));
    end
end

%%% Calculate Intermediate Values to be used in the Corrector Part
u_p = U_p./h_p;    % backing out u
v_p = V_p./h_p;    % backing out v

S_p = v_p.^2.*h_p + 0.5*g*h_p.^2;
G_p = u_p.*v_p.*h_p;
Sfy_p = n^2*v_p.*sqrt(u_p.^2 + v_p.^2)./h_p.^(4/3);

%%% The corrector step (forward difference)

% % Calculate the Boundary values
% % For the CLOSED Boundary condition. Always notice the sign changes in V and
% %G but not S
%
% for i = 1:NRows
%     for j = NCols
%         h_c(i,j) = 0.5 * (h_o(i,j) + h_p(i,j) - (del_t/grid_size).*(-V_p(i,j-1)-V_p(i,j)) +
del_t.*(R(i,j)-I(i,j)));

```



```

%    U_c(i,j) = 0.5 * (U_o(i,j) + U_p(i,j) - (del_t/grid_size).*(-G_p(i,j-1)-G_p(i,j)));
%    V_c(i,j) = 0.5 * (V_o(i,j) + V_p(i,j) - (del_t/grid_size).*(S_p(i,j-1)-S_p(i,j))+
del_t*g*0.5*(h_p(i,j-1) + h_p(i,j)).*(-(Z(i,j-1) - Z(i,j))/grid_size - Sfy_p(i,j)));
% end
% end

% For the OPEN Boundary condition, no sign changes.
for i = 1:NRows
    for j = NCols
        h_c(i,j) = 0.5 * (h_o(i,j) + h_p(i,j) - (del_t/grid_size).*(V_p(i,j-1)-V_p(i,j)) + del_t.*(R(i,j)-
I(i,j)));
        U_c(i,j) = 0.5 * (U_o(i,j) + U_p(i,j) - (del_t/grid_size).*(G_p(i,j-1)-G_p(i,j)));
        V_c(i,j) = 0.5 * (V_o(i,j) + V_p(i,j) - (del_t/grid_size).*(S_p(i,j-1)-S_p(i,j))+
del_t*g*0.5*(h_p(i,j-1) + h_p(i,j)).*(-(Z(i,j-1) - Z(i,j))/grid_size - Sfy_p(i,j)));
    end
end
% Pre-allocate with boundary values.
%Pre-allocating here fixes the size of the matrices and ensures they don't
%change once they get inside the for loops below. If I don't pre-allocate
%here, the resulting matrices from the for loops below would have a size of
%[NRows, NCols - 1] instead of the normal [NRows, NCols]. I am using the
%last column (j = NCols)to pre-allocate since that is the only known "corrected" value so
%far.

h_c = repmat(h_c(:,NCols),1,NCols); % fill up with matrix with this boundary value
U_c = repmat(U_c(:,NCols),1,NCols);
V_c = repmat(V_c(:,NCols),1,NCols);

for i = 1:NRows
    for j = 1:NCols-1
        h_c(i,j) = 0.5 * (h_o(i,j) + h_p(i,j) - (del_t/grid_size).*(V_p(i,j+1)-V_p(i,j)) + del_t.*(R(i,j)-
I(i,j)));
        U_c(i,j) = 0.5 * (U_o(i,j) + U_p(i,j) - (del_t/grid_size).*(G_p(i,j+1)-G_p(i,j)));
        V_c(i,j) = 0.5 * (V_o(i,j) + V_p(i,j) - (del_t/grid_size).*(S_p(i,j+1)-S_p(i,j))+
del_t*g*0.5*(h_p(i,j+1) + h_p(i,j)).*(-(Z(i,j+1)-Z(i,j))/grid_size - Sfy_p(i,j)));
    end
end

u_c = U_c./h_c;    % backing out u
v_c = V_c./h_c;    % backing out v
%% Computing variables to be used in handling steep gradients (see the note in the
"artificialviscosity" file for details):

[eps_back_y_hc, eps_for_y_hc] = artificialviscosity2D_y(h_c,k);
% [eps_back_x_uc, eps_for_x_uc] = artificialviscosity2D(u_c,k);
% [eps_back_x_vc, eps_for_x_vc] = artificialviscosity2D(v_c,k);

```

% Update the INTERIOR values of h,u and v:

```
for i = 1:NRows
    for j = 2:NCols-1
        h_c(i,j) = h_c(i,j) + eps_for_y_hc(i,j).*(h_c(i,j+1) - h_c(i,j)) - eps_back_y_hc(i,j).*(h_c(i,j) - h_c(i,j-1));
        % u_c(i,j) = u_c(i,j) + eps_for_x_uc(i,j).*(u_c(i+1,j) - u_c(i,j)) - eps_back_x_uc(i,j).*(u_c(i,j) - u_c(i-1,j));
        % v_c(i,j) = v_c(i,j) + eps_for_x_vc(i,j).*(v_c(i+1,j) - v_c(i,j)) - eps_back_x_vc(i,j).*(v_c(i,j) - v_c(i-1,j));
    end
end
end
```

```
function[eps_back_x, eps_for_x] = artificialviscosity2D_x(h_o,k)
```

```
%{
```

ARTIFICIAL VISCOSITY COMPUTATION 2D Saint Venant Equation for the x-direction

Jameson, et al. 1981 developed a procedure to dampen the high frequency oscillations observed near steep gradients.

These oscillations are produced as a result of dispersive errors in the MacCormack Scheme.

Details about this procedure can be found in Chaudhry (1993), Chapter 8: Computation of Rapidly Varied Flows, In:

"Open-Channel Flow". It is strongly recommended that this reference chapter in the book be read before attempting

to use - or debug - this function file.

Please keep in mind that changes in the x-Cartesian coordinates affect columns (y) in MATLAB while changes along the y-Cartesian coordinates affect rows(x) in MATLAB. This dichotomy between the Cartesian coordinate system and how MATLAB interprets rows and columns is a common cause of confusion when computing gradients in more than one-dimension.

```
%}
```

```
%% Pre-process Data
```

```
h_visc = h_o; % Set Initial value of testing array h_visc from input array h_o provided.
```

```
[NRows NCols] = size(h_o);
```

```
viscos_x = zeros([NRows NCols]); % Pre-allocate the viscosity with zeros
```

```
%% Viscosity Calculation
```

```

% For points where (i-1,j) doesn't exist, the viscosity is calculated as:
for i = 1
    for j = 1:NCols
        viscos_x(i,j) = abs(h_visc(i+1,j)- h_visc(i,j))/(abs(h_visc(i+1,j))+ abs(h_visc(i,j)));
    end
end

```

```

% For points where (i+1,j) doesn't exist, the viscosity is calculated as:
for i = NRows
    for j = 1:NCols
        viscos_x(i,j) = abs(h_visc(i,j)- h_visc(i-1,j))/(abs(h_visc(i,j))+ abs(h_visc(i-1,j)));
    end
end

```

```

% For all other interior points, the viscosity is calculated as:
for i = 2:NRows-1
    for j = 1:NCols
        viscos_x(i,j) = abs(h_visc(i+1,j)- 2*h_visc(i,j)+ h_visc(i-1,j))/(abs(h_visc(i+1,j))+
abs(2*h_visc(i,j))+ abs(h_visc(i-1,j)));
    end
end

```

```

% This is like a boundary condition of some sort since there's no "zeroth" element to
compare the first one with
for i = 1
    for j = 1:NCols
        eps_back_x(i,j) = k*viscos_x(i,j);
    end
end

```

```

% For all other interior points, the epsilon is calculated as:
for i = 2:NRows
    for j = 1:NCols
        eps_back_x(i,j) = k*max(viscos_x(i-1,j), viscos_x(i,j));
    end
end

```

```

% This is like a boundary condition of some sort since there's no "outside" element to
compare the last one with
for i = NRows
    for j = 1:NCols
        eps_for_x(i,j) = k*viscos_x(i,j);
    end
end

```

```

for i = 1:NRows-1

```

```

    for j = 1:NCols
        eps_for_x(i,j) = k*max(viscos_x(i,j), viscos_x(i+1,j));
    end
end

```

```

function[eps_back_x, eps_for_x] = artificialviscosity2D_x(h_o,k)

```

```

%{
ARTIFICIAL VISCOSITY COMPUTATION 2D Saint Venant Equation for the
x-direction
Jameson, et al. 1981 developed a procedure to dampen the high frequency oscillations
observed near steep gradients.
These oscillations are produced as a result of dispersive errors in the MacCormack
Scheme.
Details about this procedure can be found in Chaudhry (1993), Chapter 8: Computation of
Rapidly Varied Flows, In:
"Open-Channel Flow". It is strongly recommended that this reference chapter in the book
be read before attempting
to use - or debug - this function file.

```

```

Please keep in mind that changes in the x-Cartesian coordinates affect columns (y) in
MATLAB while changes along the
y-Cartesian coordinates affect rows(x) in MATLAB. This dichotomy between the Cartesian
coordinate system and how MATLAB
interprets rows and columns is a common cause of confusion when computing gradients in
more than one-dimension.

```

```

%}

%% Pre-process Data
h_visc = h_o; % Set Initial value of testing array h_visc from input array h_o provided.
[NRows NCols] = size(h_o);
viscos_x = zeros([NRows NCols]); % Pre-allocate the viscosity with zeros

```

```

%% Viscosity Calculation

```

```

% For points where (i-1,j) doesn't exist, the viscosity is calculated as:
for i = 1
    for j = 1:NCols
        viscos_x(i,j) = abs(h_visc(i+1,j)- h_visc(i,j))/(abs(h_visc(i+1,j))+ abs(h_visc(i,j)));
    end
end

```

```

% For points where (i+1,j) doesn't exist, the viscosity is calculated as:
for i = NRows
    for j = 1:NCols
        viscos_x(i,j) = abs(h_visc(i,j)- h_visc(i-1,j))/(abs(h_visc(i,j))+ abs(h_visc(i-1,j)));
    end
end

```

```

    end
end

% For all other interior points, the viscosity is calculated as:
for i = 2:NRows-1
    for j = 1:NCols
        viscos_x(i,j) = abs(h_visc(i+1,j)- 2*h_visc(i,j)+ h_visc(i-1,j))/(abs(h_visc(i+1,j))+
abs(2*h_visc(i,j))+ abs(h_visc(i-1,j)));
    end
end

% This is like a boundary condition of some sort since there's no "zeroth" element to
compare the first one with
for i = 1
    for j = 1:NCols
        eps_back_x(i,j) = k*viscos_x(i,j);
    end
end

% For all other interior points, the epsilon is calculated as:
for i = 2:NRows
    for j = 1:NCols
        eps_back_x(i,j) = k*max(viscos_x(i-1,j), viscos_x(i,j));
    end
end

% This is like a boundary condition of some sort since there's no "outside" element to
compare the last one with
for i = NRows
    for j = 1:NCols
        eps_for_x(i,j) = k*viscos_x(i,j);
    end
end

for i = 1:NRows-1
    for j = 1:NCols
        eps_for_x(i,j) = k*max(viscos_x(i,j), viscos_x(i+1,j));
    end
end

```

Appendix F

Procedure on how to Display the Quiver Plot when the number of cells is very large

%{

The purpose of this script file is to show a neat way to display the quiver plot of the velocity field when the number of elements is large.

If the number of elements in a matrix is large (>1000), the arrows tend to be small and the directional information that a quiver plot is meant to show is lost. This program averages the values of neighboring cells so that the length of each dimension of a matrix is cut into two, four, eight etc. This will correspondingly reduce the number of elements exponentially.

Lets say I have a (20,30) matrix that I want to reduce to a (10,15) by averaging values of neighboring cells. Note that this will reduce the number of elements from 600 to 150. It is much clearer to display 150 arrows than 600 in a quiver plot! The procedure for the problem described is:

```
X = rand(20, 30); % X is my matrix, a 2-D array
R = reshape(X, 2, 10, 2, 15); % this turns X into a 4-dimensional array (Note that 2*10 = 20
and 2*15 = 30)
S = sum(sum(R, 1), 3) * 0.25; % sum along the 1st and then 3rd dimensions then take the
average by dividing by 4 (or multiply by 0.25)
Y = reshape(S, 10, 15); % Reshape into the size you want i.e. a (10,15)matrix
```

This can be written in one line as:

```
Y = reshape(sum(sum(reshape(rand(20,30), 2, 10, 2, 15), 1), 3) * 0.25, 10, 15); % Cool, isn't
it? :)
```

If the matrix you are working with is big and you need to reduce by, say, 8 (which will cut the number of elements down by 64) or 16 (which will reduce the number of elements by 256), a one-line command can achieve this also.

Let's assume you are working with a (400,800):

```
Y = reshape(sum(sum(reshape(rand(400,800), 8, 50, 8, 100), 1), 3) * 0.125, 50, 100);
% Note that 1, 8*50 = 400 and 8*100 = 800; 2, You now multiply by 0.125 (or divide by 8,
whichever you want)
```

To reduce each dimension by a factor of 16:

```
Y = reshape(sum(sum(reshape(rand(400,800), 16, 25, 16, 50), 1), 3) * 0.0625,25, 50);  
% Note that 1, 16*25 = 400 and 16*50 = 800; 2, You now multiply by 0.0625 (or divide by  
16, whichever you want)
```

You should get the gist by now.

```
%}
```

%For our St. Venant's equations, a (50,40) matrix can be reduced to a (25,20) matrix
similarly:

```
udisplay_sum = reshape(sum(sum(reshape(udisplay, 2, 25, 2, 20), 1), 3) * 0.25, 25, 20);  
vdisplay_sum = reshape(sum(sum(reshape(vdisplay, 2, 25, 2, 20), 1), 3) * 0.25, 25, 20);  
quiver(udisplay_sum,vdisplay_sum) %this quiver is much clearer than if we didn't reduce  
the matrix
```

Appendix G

Determining Soil Erodibility (k_t) and Critical Shear Stress (τ_c) values for a given set of soil conditions

Soil erodibility and critical shear stress are two of the most important property values required for physically-based soil erosion modeling. Erodibility (k_t) is a soil property that quantitatively describes the erosion potential of a particular soil. Erosion occurs when a certain amount of shear stress needed to cause particle removal from the sides of a channel is reached. The critical shear stress (τ_c) is the shear stress threshold at which hydraulic forces will begin to remove significant amounts of bank material. According to Nearing et al (1989a), these two properties are related to the erosion rate by the equation:

$$D_r = K_t (\tau - \tau_c) \left(1 - \frac{qc}{T_c}\right)$$

where D_r is rill detachment rate, $\text{kg m}^{-2} \text{s}^{-1}$;

τ is the shear stress of flowing water, N m^{-2} ;

τ_c is the critical shear stress of soil, N m^{-2} ;

q is unit flow rate, $\text{m}^3 \text{s}^{-1} \text{m}^{-1}$; and c is sediment concentration, kg m^{-3} ;

T_c is transport capacity of the flowing water, $\text{kg s}^{-1} \text{m}^{-1}$; and

K_t is the erodibility of soil, s m^{-1} .

This equation is made use of in the WEPP erosion model. More recent works by Hanson (1990a, b) however present a more simplified form which is used in erosion models such as HEC-6 and SWAT

$$\varepsilon = K_t (\tau_a - \tau_c)^a$$

Where ε = erosion rate (m/s)

k_t = erodibility coefficient (m³/N·s)

a = exponent typically assumed to be 1

τ_a = applied shear stress on the soil boundary (Pa)

τ_c = critical shear stress (Pa).

Critical shear stress

Critical shear stress of a soil is usually determined in flume studies. The estimation is based on soil parameters such as particle size and soil specific gravity. The empirical relations developed by Smerdon and Beasley (1961) are presented here:

$$\tau_c = 0.16(I_w)^{0.84}$$

$$\tau_c = 10.2(D_r)^{-0.63}$$

$$\tau_c = 3.54 \times 10^{-28.1D_{50}}$$

$$\tau_c = 0.493 \times 10^{0.0182P_c}$$

Where I_w = plasticity index

D_r = dispersion ratio

D_{50} = mean particle size (m)

P_c = percent clay by weight (%).

The relations with I_w and D_r were considered the most reliable because the two parameters are directly related to cohesion properties of the soil.

Soil erodibility

Simple relations between k_t and soil properties are not readily available so one relies on empirical relations between k_t and τ_c for estimating erodibility. Two empirical methods exist for estimating k_t if τ_c is known, the Osmorne and Thorne (1988) approach which calculated the erodibility as the initial lateral bank erosion (dB) divided by τ_c , the value of dB is specified by the relation:

$$dB = \frac{223 \times 10^{-4} \tau_c e^{-0.13\tau_c}}{\gamma}$$

Where dB = initial lateral bank erosion rate (m/min per unit area)

τ_c = critical shear stress (dynes/cm²)

γ = soil unit weight (kN/m³).

The second approach utilizes the work of Hanson and Simon (2001), in which they detected an inverse relationship between the two properties given by:

$$k_t = 0.2\tau_c^{-0.5}$$

Where k_t is the erodibility coefficient (cm³/N·s).