Copyright

by

Jaehak Jeong

2008

The Dissertation Committee for Jaehak Jeong certifies that this is the approved version of the following dissertation:

A Hydrodynamic Diffusion Wave Model for Stormwater Runoff on Highway Surfaces at Superelevation Transitions

Committee:

Randall J. Charbeneau, Supervisor

David R. Maidment

Ben R. Hodges

Michael Barrett

Kamy Sepehrnoori

Lance Manuel

A Hydrodynamic Diffusion Wave Model for Stormwater Runoff on Highway Surfaces at Superelevation Transitions

by

Jaehak Jeong, B.E.; M.S.

Dissertation

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

The University of Texas at Austin
May 2008

Dedication

To Kyeong-Hwa and Hansol (Emily)

Acknowledgements

I am immensely indebted to my advisor Dr. Randall J. Charbeneau for giving me the opportunity to be part of his research group and showing the passion in the search for meaningful ways to better promote science in engineering practice by putting the best forces at work. Special thanks go to my committee members Dr. David R. Maidment, Dr. Ben Hodges, Dr. Mike Barrett, Dr. Kamy Sepehrnoori, and Dr. Lance Manuel for their valuable comments, recommendations and contributions.

I want to give my thanks and special recognition to Texas Department of Transportation for funding and supporting my Ph.D. research program at the University of Texas at Austin. I gratefully acknowledge the help of Dr. Thomas Rioux who is indispensable in the process of learning the GEOPAK interfaces.

Finally, to all the friends in Charbeneau Research group for their permanent academic and personal advice. In particular, I would like to thank to Julien Villard, Lauren Schneider, Emily Reeder, and Wa Seong (Andy) Chan whose collective effort enabled our group to work as a team in a harmonious environment. Special thanks to Dr. Masatsugu Takamatsu, a former Ph.D. student, for sharing invaluable knowledge and inspiration.

A Hydrodynamic Diffusion Wave Model for Stormwater Runoff on Highway Surfaces at Superelevation Transitions

Publication No.__________________

Jaehak Jeong, Ph.D.
The University of Texas at Austin, 2008

Supervisor: Randall J. Charbeneau

Superelevation transition is often used to help balance the centrifugal forces on vehicles through curved roadway sections. Such transitions have regions with near-zero cross-slope as the pavement cross-section rotates from a negative to positive grade. For drainage of roadway surfaces, regions with near-zero slope constitute 'irregular topography'. This condition promotes extended stormwater runoff drainage path lengths and may result in excessive splash from vehicles and hydroplaning. A critical concern is the effect of longitudinal slope on stormwater drainage through superelevation transition. The overall goal of this study is to provide design guidance on longitudinal slope at superelevation transitions through application of a numerical simulation model of highway drainage.

Sheet flow on urban pavement surfaces is very shallow, typically measuring a depth less than one centimeter. For modeling of such flow conditions, any small discontinuity or over-simplification of the surface geometry may result in failure in the flow computation. The kinematic wave approximation to the full Saint-Venant equations is often used in many surface and subsurface water models due to its simplicity in application. However, this model fails when backwater effects, ponding, or flow on reverse slope occurs in the local scale. Furthermore, due to the complexity in the surface geometry and the existence of drainage systems, the kinematic wave model is not sufficient for modeling urban stormwater runoff. On the other hand, the full dynamic wave (DW) model usually requires more computational effort. The long computation time of DW model often compromises the accuracy of the model, making the model practically inefficient.

In this study, an algorithm was developed to properly represent the irregularly shaped roadway surfaces near superelevation transition areas with unevenly spaced curvilinear grids based on the geometry profile provided by a roadway design software package such as MicroStation CAD. With this accurately defined geometric representation, a nonlinear hydrodynamic diffusion wave model for hydraulic analysis developed in this research estimates the flow depth and runoff volume on the pavement surfaces. The model computes the flow responses for rising hydrographs using a preconditioned general Conjugate Gradient method. Kinematic boundary conditions developed for the open boundaries at the upstream and downstream boundaries compute the boundary values explicitly at each time step. The result of a numerical experiment shows that the spread and concentration of sheet flow is closely related to the transition in cross slope, longitudinal slope, rainfall intensity, and the width of the road. The characteristics of the sheet flow on superelevation transition areas are analyzed to find the optimal longitudinal slope. It is found that the longitudinal slope in the range of 0.3%-0.4% is the optimal slope at superelevation transition areas which minimizes the depth of stormwater runoff. An example application of the model on a rural highway in Texas is also presented. It is found that a significant amount of stormwater may exist on traffic lanes at the superelevation transitions tested. The predicted ponding depth exceeds the minimum value for potential hydroplaning, and the pattern of the flow concentration may cause differential drag forces on traffic vehicles.

Table of Contents

Chapter 1: Introduction..1

1.1 Background..1

1.1.1 Superelevation Transition..1

1.1.2 Drainage near Zero Cross Slope Section.....................................3

1.1.3 Experimental Study...5

1.2 Motivation...8

1.3 Research Objectives...10

1.4 Organization of thr Dissertation...12

Chapter 2: Literature Review...13

2.1 Geometric Representation..13

2.2 Superelevation Transitions..14

2.3 Surface Runoff Models..16

2.3 Summary...23

Chapter 3: Grid Generation..24

3.1 Introduction...24

3.2 Geometry Data from GEOPAK...24

3.3 Curvature Geometry..25

3.4 Grid Generation for Curvature..28

3.5 Characterization of the Parametric Mapping...................................31

3.6 Algorithm for Linear Section..35

3.7 Geometry Data Screening...37

3.8 Restrictions on the Geometry of Roadway.....................................39

Chapter 4: Model Development...41

4.1 Introduction...41

4.2 Saint Venant Equations for Sheet Flow..41

4.3 Diffusion Wave Model with Manning’s Equation..........................43

4.4 Numerical Model Development...45

4.5 Solution Process..51

4.6 Stability 55
4.7 Evaluation of Model Convergence 58
4.8 Model Verification...65
4.9 Discussion on Flow Models...68
4.10 Initial Condition and Boundary Conditions...72
    4.10.1 Initial Condition...72
    4.10.2 Kinematic Boundary Conditions 72
    4.10.3 No-flow (Neumann) Boundary Conditions 78
4.11 Algorithm for Curb-opening Inlets 79

Chapter 5: Numerical Experiment 85
5.1 Description..85
5.2 Analysis...87
    5.2.1 *Type-I* 87
    5.2.2 *Type-II*..91
5.3 Sensitivity Test...95
    5.3.1 Longitudinal Slope..95
    5.3.2 Rainfall Intensity...101
    5.3.3 Surface Roughness..103
    5.3.4 Number of Traffic Lanes 104
    5.3.5 Location of Curb-opening Inlets 108
5.4 Residence Time of Stormwater Runoff 109
5.5 Discussion 112

Chapter 6: Model Application 114
6.1 Introduction..114
6.2 Result and Discussion 116

Chapter 7: Conclusions and Recommendations for Future Work 121
7.1 Conclusions..121
7.2 Recommendations for Future Work...125

Appendix A: Algorithm for Iterative Method...126
A.1 Matrix Factorization...126

A.2 General Conjugate Gradient Method 128

Appendix B: Creating Superelevation Transition in GEOPAK 131

Appendix C: FORTRAN Source Code and Output Files 149

C.1 Source code 149

C.2 Output Files 185

C.2.1 Computation Related Files 185

C.2.2 Geometry Related Files 185

C.2.3 Result Files 185

C.2.4 Example 185

References 187

Vita ……………………………………………………………………………191

List of Tables

Table 4.1: Computation time with respect to number of lanes 65

Table 4.2: Errors in the upstream boundary condition 78

Table 4.3: Errors in the downstream boundary condition.. 78

Table 5.1: Variables for linear regression of the maximum water depth w.r.t. rainfall intensity 102

Table 5.2: Manning’s n for Street and Pavement Gutters (HEC-22, 2001)................. 103

Table 5.3: Estimated difference in residence time of stormwater runoff between normal crown and superelevation transition sections 112

Table 6.1: Properties of the road segments and estimated values................................ 116

List of Figures

Figure 1.1: Superelevation transition with change in roadway cross-slope and stormwater drainage paths 2

Figure 1.2: Schematic plan view of pavement near superelevation transition, showing the drainage path with maximum length............ 4

Figure 1.3: Picture of experimental setup............ 6

Figure 1.4: Manning’s Coefficient as a function of Reynolds number for a concrete surface. Solid = Rain only, Open = No rain but stream flow only 8

Figure 2.1: Diagrammatic profiles showing methods of attaining superelevation for a Curve to the Right (Source: AASHTO Green Book (2004))............ 14

Figure 2.2: The graphic user interface for extracting 3D roadway surface profile in GEOPAK 16

Figure 2.3: Partition of the *K, F_0* field into three zones for zero-depth-gradient downstream boundary conditions. (Vieira, 1983)............ 21

Figure 3.1: An example of geometric information of a roadway in the DTM 25

Figure 3.2: Sequence of centerline points with radius of curvature $R(\xi_c)$ 26

Figure 3.3: Geometry for curvature algorithm 26

Figure 3.4: Cases where θ_i needs to be modified 29

Figure 3.5: Geometry for grid point generation algorithm............ 30

Figure 3.6: Model grid in prototype data space............ 31

Figure 3.7: Grid layout for a domain with a curved roadway 34

Figure 3.8: Geometry for grid generation for straight section of roadway 35

Figure 3.9: Use of filtering to minimize “data-noise” in roadway direction............ 38

Figure 3.10: Curvature profiles for a roadway section............ 39

Figure 4.1: Overland flow over a plane............ 42

Figure 4.2: Transformation of grid space 46

Figure 4.3: Transformed grid for cell (i,j) 47

Figure 4.4: Transformed model grid 52

Figure 4.5: Pentadiagonal matrix systems............ 53

Figure 4.6: Solution Process 55
Figure 4.7: Convergence speed of MICCG solver measured by L_2 and L_∞ norms. 59
Figure 4.8: Convergence and errors in 1D simulation 60
Figure 4.9: Errors in 2D simulation for different cell sizes 62
Figure 4.10: Error in the solution for 1D flow with different time intervals (top: linear model, bottom: nonlinear model) 63
Figure 4.11: Comparison of the model solutions with kinematic wave model solutions at different time levels. r = 250mm/hr, n = 0.015, L = 30 m, Δx = 0.6 m, N_x = 50. (S_o = 1% shown on the left, 5% on the right) 67
Figure 4.12: Rising hydrograph of the diffusion wave model for a 1D flow: S_o = 1%, n = 0.015, r = 200 mm/hr, L = 30 m. 68
Figure 4.13: 1D flow under constant rainfall at steady state 69
Figure 4.14: Kinematic wave solution for a 2D flow 71
Figure 4.15: Implementation of kinematic boundary conditions when characteristics arrive at the boundary from outside (left) of the domain and within (right) the domain 73
Figure 4.16: Incremental rainfall loading on the upstream boundary. ($S_{o\xi}$= 2%, $S_{o\eta}$= 2%, W = 7.3m, $\Delta x = \Delta y$ = 0.3m, r = 125mm/hr, n = 0.015) 75
Figure 4.17: Schematic view of computational grid for outflow kinematic boundary condition 77
Figure 4.18: A depressed curb-opening inlet (HEC-12). 80
Figure 4.19: Cell scale configuration of a curb-opening inlet 80
Figure 4.20: Schematic of the cell scale configuration of a curb-opening inlet 83
Figure 4.21: Depression in ponding depth at the inlet (node 160) 83
Figure 4.22: Performance of curb-opening inlet 84
Figure 5.1: Types of the roadway surfaces used in the numerical experiment. 85
Figure 5.2: Contour plot of the surface elevation of a *Type-I* road (4-lane, S_x=1.0%, ZCS at 122m station, elevation in feet). 86
Figure 5.3: Contour plot of the surface elevation of a *Type-II* road (4-lane road, S_x=1.0%, ZCS at 103m station, elevation in feet). 87

Figure 5.4: The profile of water depth at the steady state condition (*Type-I*, r=250mm/hr, 4-lane road, depth in millimeters). 88
Figure 5.5: Vector plots of the unit flow rate at the steady state condition (*Type-I*, r=250mm/hr, 4-lane road, depth in millimeters). 90
Figure 5.6: The profile of water depth at the steady state condition (*Type-II*, r=250mm/hr, 4-lane road, depth in millimeter). 92
Figure 5.7: Vector plots of the unit flow rate at the steady state condition (*Type-II*, r=250mm/hr, 4-lane road). 93
Figure 5.8: Longitudinal profile of ponding depth at the inside end of 8-lane road under 250mm/hr rainfall (*Type-II* roads). 94
Figure 5.9: Cross sectional profile of water depth at different locations of the *Type-I* roads shown in Figure 5.4. 96
Figure 5.10: Locations where peak depth at the steady state condition occur on various longitudinal slope surfaces (*Type-I* roads). 97
Figure 5.11: Saddle point at the ZCS section on a 0.1% slope road. Contours show surface elevation and arrows represent the direction and magnitude of unit flow on the surface at steady state. 98
Figure 5.12: Contour of the surface elevation near the stagnation point (red star) on different slopes (*Type-I* roads). 100
Figure 5.13: Linearity in the maximum ponding depth with respect to rainfall intensity (S_x=1.0%). 101
Figure 5.14: Flow responses on the downstream end of a 100 feet road surface with Sx=0.1% for different values of surface roughness coefficient. 104
Figure 5.15: Maximum ponding depths on the traffic lanes (r=250mm/hr). 105
Figure 5.16: Maximum ponding depth (r=250mm/hr). 107
Figure 5.17: Maximum ponding depths on the traffic lanes (shoulder area excluded) on *Type-I* roads (r=250mm/hr). 108
Figure 5.18: Residence time of stormwater runoff 110
Figure 5.19: Box-plot of the difference in residence time between superelevation transition and normal crown sections 111

Figure 6.1: RM 2338 in Williamson County, Texas 114
Figure 6.2: Profile of RM 2338 in the GEOPAK shape file (Dimensions in feet)........ 115
Figure 6.3: Profile of sheet flow on the road at steady state condition (r=250mm/hr, depth in millimeter).. 118
Figure 6.4: Nonexceedance probabilistic distributions of ponding depth for selected 4 segments.. 119

Nomenclature

Symbols	Definition
b_w	adjustment factor for n_l
C_i	chord length of i^{th} station (m)
c_o	mean texture depth (m)
C_ξ	conveyance in ξ direction
C_η	conveyance in η direction
$D(h)$	diffusion coefficient (m^2/s)
e_d	design superelevation rate
e_{NC}	normal cross slope rate (%)
f	side friction factor
g	gravitational acceleration ($=9.8 m/s^2$)
h	flow depth (mm)
H	total head ($=h+z_B$) (m)
h_c	critical depth (m)
J	Jacobian of transformation
L_d	drainage path length (m)
L_r	length of superelevation runoff (m)
L_t	length of tangent runout (m)
L_ξ	length scaling for segments in ξ
L_η	length scaling for segments in η
n	Manning's coefficient
N_x	number of grid cells in the x direction
N_y	number of grid cells in the y direction
n_l	number of lanes rotated
q_{max}	maximum unit discharge at T_c
q_x	unit discharge in the x direction (m^2/s)
q_y	unit discharge in the y direction (m^2/s)

Symbols	**Definition**
r	rainfall intensity (mm/hr)
R_i	radius of curvature of i^{th} station (m)
S_{fx}	friction slope in the x direction
S_{fy}	friction slope in the y direction
S_{ox}	surface slope in the x direction
S_{oy}	surface slope in the y direction
T_c	time of concentration (sec)
V	design velocity (km/hr)
v_c	critical velocity (m/s)
w	width of a traffic lane (m)
W	width of roadway
$(x_c,y_c)_i$	(x,y) coordinates of the center of road
$(x_{cc},y_{cc})_i$	(x,y) coordinates of the center of curvature
z_B	surface elevation (m)
z_{inlet}	surface elevation of inlet opening (m)

Greek Symbols	**Definition**
α	direction angle of the roadway (rad)
β	direction of curvature (rad)
Δ	maximum relative gradient (%)
Δx	grid spacing in the x direction
Δy	grid spacing in the y direction
θ	direction angle to the center of curvature (rad)
ξ	longitudinal axis in a curvilinear coordinate system
η	transverse axis in a curvilinear coordinate system

Dimensionless numbers	Definition
Re	Reynolds number $=\frac{q}{v}$
F	Froude number $=\frac{q}{\sqrt{gh^3}}$
K	Kinematic wave number $=\frac{S_o L_o}{h_o F_o{}^2}$

Abbreviation	Definition
1D	1-dimensional
2D	2-dimensional
3D	3-dimensional
B.C.	boundary condition
CDS	central differencing scheme
CFL	Courant–Friedrichs–Lewy
CN	Crank-Nicolson
DTM	digital terrain model
DW	diffusion wave
DYW	dynamic wave
FVM	finite volume method
KW	kinematic wave
MICCG	modified incomplete conjugate gradient
RMSE	root mean square error
SCS	soil conservation service
ZCS	zero cross slope

Chapter 1: Introduction

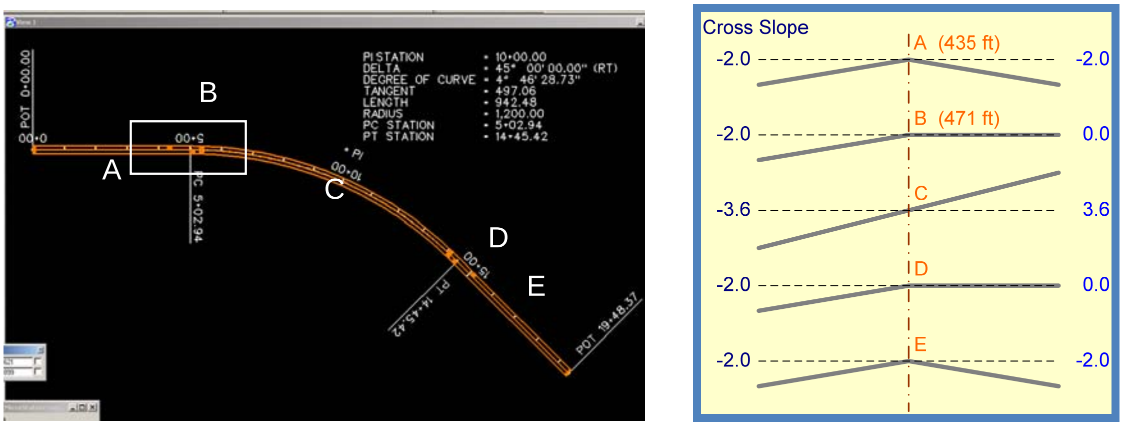
1.1 BACKGROUND

1.1.1 Superelevation Transition

The majority of highways in the US are constructed with nonporous pavement. Wet conditions on roadways remain a problem, impacting safety when excessive rainfall remains on the pavement before it drains. Natural rainfall on the pavement surface forms sheet flow which is also called water film or overland flow. Even a very thin water film may decrease the friction between tires and roadway surface promoting hydroplaning of vehicles. Thus, the presence of water slows traffic and contributes to accidents from hydroplaning and loss of visibility from splash and spray (Anderson et al., 1998). Ponding of water may also result in dangerous torque levels on vehicles and ultimate loss of vehicle control (FHWA HEC-12, 1984). Horizontal drag forces imposed on the vehicle by the water may cause hydroplaning. If the drag forces are unevenly distributed in the lateral direction by different ponding depths, it can cause hazardous directional instability (torque) to vehicles on the roadway.

As vehicles move through curved highway sections, centripetal acceleration naturally occurs toward the center of curvature. In reaction, drivers feel outward (centrifugal) forces as vehicles are pushed out in the radial direction of the curve. These forces can be controlled through use of limits on curve radius and through superelevation of traffic lanes. In traffic engineering, the degree of superelevation is determined by balancing centrifugal forces on vehicles with the side friction between tires and roadway surface. The side friction factor decreases not only when design speed increases but also when ice or ponded water exist on the pavement.

A schematic plan view of a superelevation transition is shown in Figure 1.1. In superelevation transitions leading to curved alignments, the cross slope of the inward travel lanes is lowered while the outer lanes are raised to form positive cross slopes (Section C in Figure 1.1). This means that travel lanes on the outward side of the radius of curvature must pass from a negative cross slope (Section A) to a positive cross slope (Section C). This transition in cross slope means that there must be a section of pavement with zero cross slope (Section B).



PI STATION = 10+00.00
DELTA = 45° 00' 00.00" (RT)
DEGREE OF CURVE = 4° 46' 28.73"
TANGENT = 497.06
LENGTH = 942.48
RADIUS = 1,200.00
PC STATION = 5+02.94
PT STATION = 14+45.42

Figure 1.1: Superelevation transition with change in roadway cross-slope and stormwater drainage paths

AASHTO (2004) provides design recommendations on the maximum relative gradient that limit the rate (with respect to longitudinal distance) at which the outer travel lane elevation can be raised or lowered, and the maximum relative gradient decreases with increasing design speed noting that a vehicle may slide inward of the curve at low speed or when ice or water exists on the pavement. This means that for highway sections with larger design speeds, the length of highway pavement with near-zero cross slope will increase. To prevent ponding of water from rainfall, the pavement in these segments with near-zero cross slope must maintain a longitudinal slope. On the other hand, through roadway segments with near-zero cross slope, increasing the longitudinal slope will increase the drainage path length and could result in increased ponding depths over parts of the roadway surface.

1.1.2 Drainage near Zero Cross Slope Section

The outside half of the roadway near the zero cross slope section (section B) in Figure 1.1 is shown in Figure 1.2 with the maximum stormwater runoff drainage path. This figure demonstrates how drainage paths increase with the existence of zero cross slope areas on the highway pavement. In this figure, it is assumed that the highway crown serves as the axis of rotation for the warped section leading to the superelevation highway section (the curvature of the highway is not shown). The elevation of the Travel Lane Edge increases along the highway longitudinal length, but at a limited rate. The pavement cross slope changes from negative (outward) to positive near the entrance of the transition. Normally a drainage inlet is located immediately upslope of the zero cross slope station to capture stormwater runoff; otherwise such runoff would spread across the pavement surface through the transition section.

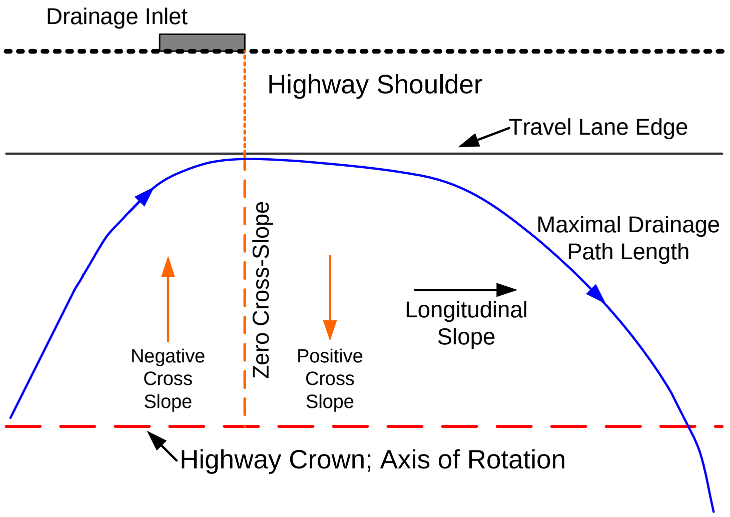


Figure 1.2: Schematic plan view of pavement near superelevation transition, showing the drainage path with maximum length.

The drainage path originates near the highway crown at a location upslope from the zero cross slope station. Because of the negative cross slope, the drainage path is initially directed towards the outer travel lane edge. However, the cross slope superelevation results in the drainage path turning inward to the inside pavement edge of the roadway curvature. As shown in Figure 1.2, the path of the maximum drainage path length becomes tangent to the outer travel lane edge near the station with zero-cross slope since cross slope is zero and longitudinal slope is non-zero at the zero cross slope section. The drainage path will cross the traffic lanes again and will also cross the traffic lanes on the inside of the transition. The direction of the drainage path is "down slope" because gravity is the primary force causing overland flow. Increasing the longitudinal slope will increase the drainage path length by increasing the length in the longitudinal direction. Rainfall will increase the discharge along the path length, and may result in increased ponding along pavement surfaces. On the other hand, decreasing the longitudinal slope will result in a shorter path length with smaller drainage rates. The smaller drainage rates increase the pavement drainage time, and may result in increased ponding depths due to continued addition of rainfall. Thus it appears that for a given rainfall intensity, there may be an “optimal” longitudinal pavement slope leading to and from superelevation transitions. Here, “optimal” may refer to controlling the maximum ponding depth, or the size of the ponding region, or other factors, under design rainfall conditions

1.1.3 Experimental Study

This research is part of a TxDOT-sponsored research program. The overall objective of this research project is to develop design guidance on longitudinal grade through superelevation transitions. An experimental study of sheet flow characteristics on rough surfaces under simulated rainfall conditions has also been conducted by Charbeneau et al. (2007). A large-scale (13 meter) experimental facility was constructed at the Pickle Research Campus. Villard (2005) describes the details of the construction of the experimental equipment. A rainfall simulator, with its unique rotating system, provides uniform rainfall on the roadway model surface. For rainfall simulation, a system of nozzles spaced at 1 meter interval spray water at an angle of 60 degrees from the nozzle centerline. To enhance the spatial uniformity of rainfall, the entire rainfall simulating system rotates at 0.25 rev/sec with a 30-cm rotation arm. This configuration was selected by first measuring the spray intensity beneath an individual nozzle, then evaluating the average rainfall intensity that would occur beneath a system of nozzles being rotated with an arm of given radius.

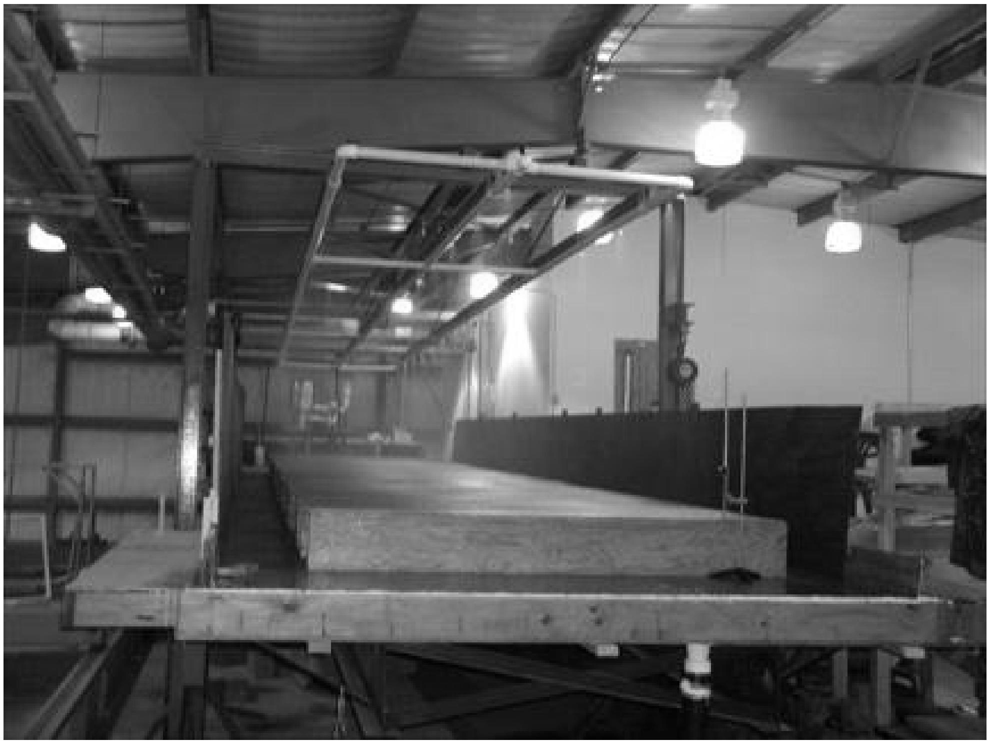


Figure 1.3: Picture of experimental setup

The roadway model is coated by resin with sand particles to simulate the roughness of asphalt and concrete type pavement surfaces. The flow measurement system consists of independent measurements of the flow depth (*h)* and unit discharge (*q)*. Water depth and flow rate were measured at the downstream end for various conditions such as different rainfall intensity, surface textures, lateral inflow from upstream, and surface slopes. The flow depth is measured using a piezometer tap located on the roadway surface and attached through tubing to an inclined monometer board. In addition to measuring flow depth using the manometer board, depth was also measured directly using a thin graduated pin acting as a small-scale ruler to evaluate measurement sensitivity to the diameter of the piezometer tap opening on the roadway surface (Schneider, 2006). The unit discharge is measured by direct capture of discharge from the roadway surface. The Manning's equation is retained for friction slope terms, S_{fx} and S_{fy}

$$q_x = \frac{1}{n} h^{5/3} S_{fx}^{\ \ 1/2} \tag{1.1}$$

where q_x is unit flow rate in the x direction, n is the Manning's coefficient, and h is water depth. As the sheet flow on the highway pavement surface is very shallow compared to the roughness thickness or the spread of flow over the plane, standard values of Manning's coefficient are not directly applicable to the numerical modeling. In the experimental study conducted at the Center for Research in Water Resources (CRWR) in the Pickle Research Campus of the University of Texas at Austin, a data set of 1,294 measurements were collected and analyzed to find an empirical definition of the Manning's coefficient for sheet flow on non-porous pavement surfaces in terms of the flow variables such as flow rate, water depth, and mean texture depth (Reeder, 2007).

$$n = 2.427 c_0 + 0.0047 \tag{1.2}$$

where c_0 denotes the mean texture depth (ASTM, 2006) in meters. Equation 1.2 is applicable to all flow regimes: laminar, turbulent, and transition flows. Therefore, the proposed empirical model for the Manning's coefficient is independent of the Reynolds number but is a function of the mean texture depth. In analyzing the collected data, a large variation is found in Manning's coefficient at low Reynolds number (see Figure 1.4) in averaging out the coefficient value to be constant. Even though the variation of Manning's coefficient at low Reynolds numbers increases uncertainty, the characteristics of sheet flow on impervious pavement surfaces represented by Equation 1.2 are incorporated in the numerical model for improving the model accuracy.

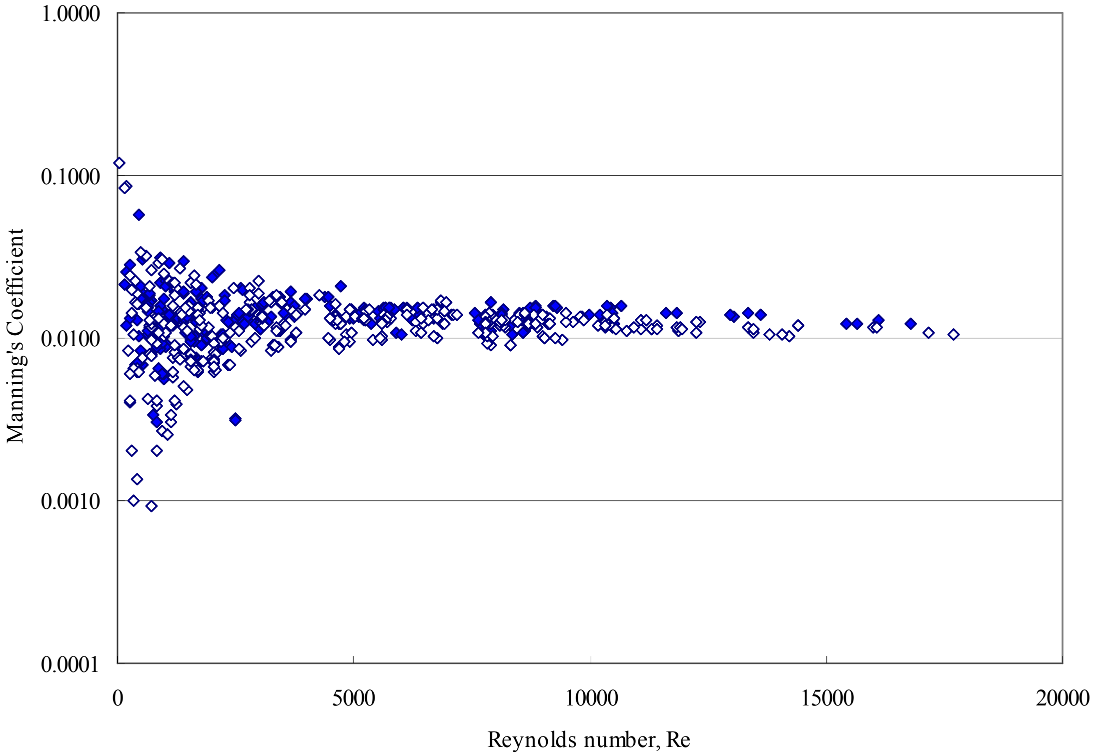


Figure 1.4: Manning's Coefficient as a function of Reynolds number for a concrete surface. Solid = Rain only, Open = No rain but stream flow only

1.2 MOTIVATION

A computer model is needed for estimating the discharge and depth of the sheet flow on highway pavement surfaces to find an optimal longitudinal slope and location of drainage inlets for any given section of roadway where zero cross slope occurs. Due to the complexity in the geometry of the road surface, the model needs to be capable of estimating flow responses anywhere on the surface. The location and the magnitude of the maximum ponding depth on traffic lanes should be quantified, allowing optimum placement of drainage inlets to minimize the spread of water on the surface.

Sheet flow is very sensitive to the geometry of pavement surfaces. Thus, the representation of the roadway topography is essential in this research. For design of all roadways, TxDOT requires use of the GEOPAK software package that runs on MicroStation CAD software. Once a roadway is designed in GEOPAK, the roadway profile can be downloaded in ASCII file format. This data set includes a series of 3D coordinates (x,y,z) of end-of-lanes including the center and side ends of the road at stations which are regularly spaced along the longitudinal direction of the roadway. These 3D locations are used to build a precise representation of the roadway geometry.

Calculation of water depth on pavement surfaces is based on overland flow routing techniques. Shallow water equations for these techniques were first published by Saint-Venant in 1848 (Chow, 1959) to describe flow in river channels. These equations constitute the continuity and dynamic equations for gradually varied unsteady flow. The full Saint-Venant equations are, however, too complicated to be solved analytically. Two simplified models of the full dynamic equations are the kinematic wave model and the diffusion wave model. For most hydrologically significant cases, the kinematic wave model provides a good approximation to the Saint-Venant equations for overland flow (Woolhiser and Liggett, 1967). However, the geometry of the highway surface near superelevation transition is complex for a flow domain. The existence of superelevation transition may induce laterally discontinuous drainage path lengths depending on how far one proceeds upstream. The kinematic wave model, which forces water to flow "down-slope" may not be appropriate for sheet flow simulation on highway pavement surfaces. The driving force in the diffusion wave model includes the depth gradient term as well as the bed slope. Compared to the full dynamic wave models, diffusion wave approximation is more efficient in terms of computational cost, as it does not have convective acceleration terms. Also, for roadway surfaces near superelevation transitions, the convective acceleration terms are not expected to be significant. Therefore, the proposed rainfall-runoff model will be developed based on diffusion wave approximation.

The following questions will be answered throughout the research with numerical experiments and model applications.

1. What is the optimal longitudinal slope that minimizes stormwater runoff on the road surface near the location of zero-cross slope?
2. Is the concentration and spread of sheet flow predictable?
3. How does the sheet flow respond to different magnitudes of rainfall intensity?
4. Should a curb-inlet be installed at the zero-cross slope section?

1.3 RESEARCH OBJECTIVES

This research project addresses issues associated with highway drainage from roadway segments leading to and from superelevation sections where pavement cross-slope is small. The amount of ponding on pavement surfaces in areas of transition from normal crown to fully-superelevated roadway sections depends on longitudinal slope and other factors. This relationship is investigated, and design guidance on longitudinal grade is provided. In addition, design methods for locating drainage inlets must be modified for superelevation transitions, and design guidance for inlet spacing is also developed. The specific objectives are listed below.

1. Develop an algorithm for representation of roadway surface geometry. The GEOPAK data file has a unique format in which the geometric data is described. The proposed model is capable of reading the unique format of the

GEOPAK data file and generating a grid space in a local curvilinear coordinate system.

2. Develop boundary conditions for open boundaries. Upstream and downstream boundaries are open to flow coming in and out. The inflow over the upstream boundary is modeled using a kinematic wave approximation. The downstream boundary condition has both a no through-flow condition and an explicit kinematic approximation condition. The development of gutter flow along the low side curb can also be simulated without special treatment in the model.

3. Develop a mathematical/numerical model for sheet flow. The diffusion wave approximation equation with nonlinear diffusion coefficient results in a 2D nonlinear parabolic partial differential equation. The convergence and stability as well as computation cost are major issues that make solving this problem difficult (Anderson et al., 1998). A General Conjugate Gradient (GCG) solver preconditioned with the Cholesky decomposition technique that is superior to other relaxation methods in convergence speed and stability (Ferziger and Peric, 2002) is built to solve the resulting systems of linear equations arising from discretization of governing partial differential equations.

4. Develop an algorithm for inlet placement. Hydraulics for curb-side inlets is reviewed to define the flow around the inlets. An algorithm that updates the information regarding the inlet placement will be built and the flow responses will be evaluated.

5. Build a FORTRAN code to carry out the computations. Because FORTRAN is the programming language of choice in the science and engineering field, mathematical/numerical models with initial/boundary conditions are incorporated in a compiled computer program written in FORTRAN 90/95 with the Lahey compiler v5.5.

1.4 ORGANIZATION OF THR DISSERTATION

There are five chapters in this dissertation. A literature review is discussed in Chapter 2. Chapter 3 describes an interface model that reads geometry information from geometry files exported from MicroStation GEOPAK. The development of physical grid space based on the DTM is also presented in this chapter. Chapter 4 describes the mathematical and numerical model. Sheet flow is defined in a mathematical expression based on the diffusion wave model theory. Open boundary conditions are explicitly defined based on a kinematic wave approximation. A Finite Volume-based set of equations is developed with which the flow depth will be estimated. The convergence speed of the model and model error are investigated for 1D and 2D problems. In Chapter 5, a set of numerical experiments is conducted for better understanding of sheet flow. Flow responses on various conditions are simulated to find the optimal longitudinal slope. The proposed numerical model is applied to an existing rural highway in Texas in Chapter 6. Finally, Chapter 7 provides discussions and concluding remarks on the research.

Chapter 2: Literature Review

2.1 GEOMETRIC REPRESENTATION

A natural watershed is extremely complex; heterogeneous in surface geometry and characteristics. As a part of hydrologic or hydraulic modeling, simplification of the surface shape and other geomorphologic characteristics of the watershed is necessary. Even though urban highways are artificially designed so they are generally not as complex as natural watersheds, the geometry of the pavement surfaces of the highway near zero-cross-slope section is irregularly complex. Since the water depth of sheet flow on roadways is usually very shallow, it is expected that the sheet flow modeling is very sensitive to the computational grid space which represents the real roadway shape. Therefore, the accuracy of the model is highly dependant on the geometric representation of the roadway surface.

Woolhiser (1969) introduced a conceptual method in which the watershed surface is simplified to a linearly converging section: a cone shape. Similar methods such as diverging section and converging-diverging section, and V-shape geometry were developed since then and used in many hydrologic modeling works (e.g. Wooding (1966), Singh and Agiralioglu (1981a, 1981b), Agiralioglu (1981)). The main drawback of these conceptual methods is that the watershed geometry is algebraically idealized so that any local geometric heterogeneity is ignored.

Recently, with the advances in digital mapping techniques, the Geographic Information System (GIS) and digital elevation models (DEM) have become popular tools in hydrologic modeling. DEMs store information such as surface geometry and morphological characteristics in a raster format. Jain and Singh (2005) developed a DEM-based diffusion wave model. They found good agreement of the model on collected data with 20 meter grid resolution for a 17.17km^2 catchment size. The main drawback of the DEMs is that as the grid resolution increases, the amount of the corresponding data also increases. Zhang and Montgomery (1994) suggest a 10 meter grid size as a rational compromise.

2.2 SUPERELEVATION TRANSITIONS

The superelevation section consists of the superelevation runoff and tangent runout as shown in Figure 2.1. In superelevation runoff, the cross-slope of outside-lane changes from zero to full superelevation, or vice versa. The tangent runout consists of the section where the cross-slope of the outside-lane changes from the normal crown to zero.

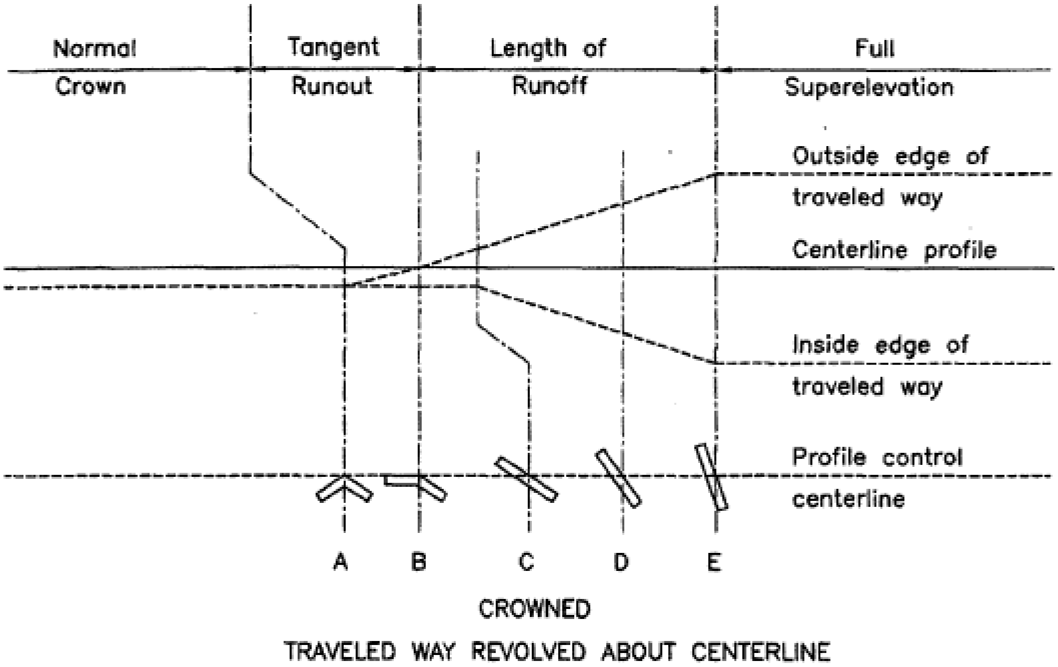

Normal Crown
Tangent Runout
Length of Runoff
Full Superelevation
Outside edge of traveled way
Centerline profile
Inside edge of traveled way
Profile control centerline
A
B
C
D
E
CROWNED
TRAVELED WAY REVOLVED ABOUT CENTERLINE

Figure 2.1: Diagrammatic profiles showing methods of attaining superelevation for a Curve to the Right (Source: AASHTO Green Book (2004)).

As a vehicle traverses a horizontal curve, the vehicle weight component due to roadway superelevation and the side friction between tires and road surface counter-balance centrifugal force. Thus, design superelevation rate is determined by design speed, side friction factor, and radius of curvature.

$$e_d + f = \frac{V^2}{127R} \tag{2.1}$$

where f is side friction factor, R is the radius of horizontal curvature [m], V is vehicle speed [km/hr], and e_d is design superelevation rate in decimal format. AASHTO Green Book (2004) suggests that the side friction factor ranges from 0.4 at 15 km/hr to 0.15 at 70 km/hr.

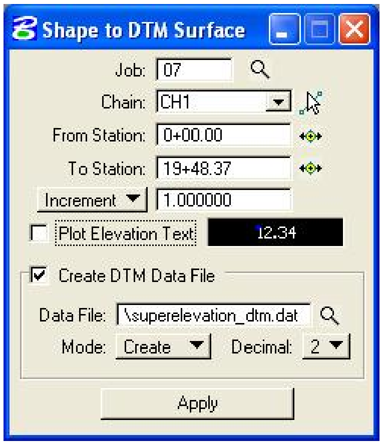
According to the AASHTO Green Book (2004), the geometry of superelevation transition areas is designed based on normal cross slope rate, design superelevation rate, the number of lanes rotated, width of a traffic lane, and adjustment factor for number of lanes rotated. The length of superelevation runoff (L_r) is based on the maximum acceptable difference (0.5%) between the longitudinal grades of the road center and the edge of pavement. The length of tangent runout (L_t) is determined by the amount of adverse cross slope to be removed and the rate at which it is removed. Based on these rationales, the following equations are used to compute the minimum values of them:

$$L_r = \frac{(wn_l)e_d}{\Delta}(b_w) \tag{2.2a}$$

$$L_t = \frac{e_{NC}}{e_d} L_r \tag{2.2b}$$

where Δ is the maximum relative gradient [%]; n_l is the number of lanes rotated; b_w is an adjustment factor for n_l; w is the width of one traffic lane [m]; e_d is the design superelevation rate [%]; e_{NC} is the normal cross slope rate [%].

The State of Texas uses the MicroStation GEOPAK software package in designing state highways. The geometric information of a designed roadway surface is stored within GEOPAK in three-dimensional coordinates. It can be exported to a Digital Terrain Model (DTM) in ASCII format text file as shown in Figure 2.2. The DTM shapes the roadway surface in three-dimensional space by specifying the point of the center and ends of lanes. Using the DTM, the sheet flow model can not only reproduce roadway surfaces efficiently, but can also be versatile in model application. Once exported in a DTM geometry file, any section of highway near a superelevation transition may be simulated by the proposed sheet flow model.



Job: 07
Chain: CH1
From Station: 0+00.00
To Station: 19+48.37
Increment: 1.000000
☐ Plot Elevation Text 12.34
☑ Create DTM Data File
Data File: \superelevation_dtm.dat
Mode: Create
Decimal: 2

Figure 2.2: The graphic user interface for extracting 3D roadway surface profile in GEOPAK

2.3 SURFACE RUNOFF MODELS

Surface runoff models are generally categorized in two groups: empirical models and hydrodynamic models. Empirical models simplify hydrologic processes by introducing empirical parameters and employing a one-dimensional treatment. The Soil Conservation Service (SCS) method developed for computing abstractions from storm rainfall has been popularly used since it was introduced in 1972. The surface runoff at the outlet of a watershed is estimated using an empirical relationship between rainfall excess and curve number which represents the degree of surface infiltration. On the other hand, the rational method is widely used for sewer design because of its simplicity (Chow et al., 1988). In this method, the rate of peak discharge, which occurs at the time of concentration, is estimated by the watershed area, rainfall intensity, and an empirical runoff coefficient which represents surface characteristics. Because the runoff coefficient is empirically determined and the nature of watershed surfaces is complex, the accuracy of the model application is heavily dependent on expertise for choosing a reasonable runoff coefficient. The unit hydrograph proposed by Sherman (1932) is a simple linear model that predicts direct runoff and stream flow hydrographs. The assumptions and limitations of this model are described by Chow et al. (1988). Empirical models are simple and easy to apply to estimate the runoff of a watershed at the outlet such as a gage station. However, the simplicity of the model makes it inapplicable for estimating the flow responses within the flow domain.

The equations of continuity and momentum for gradually varied unsteady shallow water flow are often referred to as the Saint Venant equations. Hydrodynamic models, which consist of the dynamic wave model, the diffusion wave model, and the kinematic wave model, solve the flow dynamics represented by the Saint Venant equations to estimate the runoff and flow responses in a watershed. The dynamic wave model takes into account the full Saint Venant equations for flow routing. The origin of the name “dynamic wave” is from the fact that the model includes the convective inertial terms in the momentum equation. Based on data taken from an actual river in steep alluvial terrain, Henderson (1966) proposed that on steep slopes only the surface slope terms need to be retained in the momentum equation, and on very flat slopes the bed slope and the pressure gradient terms need to be retained.

Lighthill and Whitham (1955a, 1955b) developed a theory for the kinematic wave model. They used the kinematic wave theory for flood routing in long rivers and traffic flow on long crowded roads. They showed that at the low Froude numbers appropriate for flood waves of a river, dynamic waves were rapidly attenuated and the main disturbance was carried downstream by kinematic waves. Henderson and Wooding (1964) applied the kinematic wave theory to the problem of overland flow and groundwater flow on a sloping plane. They found good agreement between the kinematic wave solution and experimental measurements for overland flow, while significant differences were found in the groundwater flow possibly due to the existence of slope of groundwater surface. They concluded that the buildup and decay of the groundwater surface made the problem of a nonlinear diffusion wave model problem. Woolhiser and Liggett (1967) applied the kinematic wave theory to model unsteady one-dimensional overland flow. They used the method of characteristics to find the flow response of the rising portion of a hydrograph. Iwagaki (1955) developed an approximation method to compute unsteady flow in open channels of any cross-sectional shape with lateral inflows using the method of characteristics. His research is restricted to rivers with steep slopes.

There are three important non-dimensional parameters that are used in hydrodynamic modeling. They are the Reynolds number (Re), the Froude number (Fr) and the kinematic wave number (K). The Reynolds number is originally defined as the ratio of inertial forces to viscous forces in pipe flows.

$$\mathrm{Re} = \frac{\rho u d}{\mu} \tag{2.3}$$

In Equation 2.3, u is the average velocity, d is the pipe diameter, ρ is the density, and μ is the dynamic viscosity of water. For cross-sections other than the pipe flows, the hydraulic radius (R_h) is used rather than the diameter (for a pipe, $R_h = d/4$). For the case of overland flow, a characteristic length scale is the flow depth (h), which is also the hydraulic radius ($h = R_h$). The Reynolds number may be defined by

$$\mathrm{Re} = \frac{q}{\nu} \tag{2.4}$$

where q is the unit discharge ($q=uh$) and ν is the kinematic viscosity ($\nu=\mu/\rho$). In Equation 2.4, the Reynolds number is defined such that the value is smaller than conventional values used for pipe flow by a factor of 4. The Reynolds number defined in Equation 2.4 is used for sheet flow modeling in this research.

The Froude number shows a characteristic of flows with the ratio of the speed of water and the celerity of disturbances.

$$F = \frac{q}{\sqrt{gh^3}} \tag{2.5}$$

The Froude number at the end of a flow path is often used in characterizing sheet flow (Liggett and Woolhiser (1967), Govindaraju et. al. (1988a, 1988b), Woolhiser and Liggett (1967)).

The kinematic wave number reflects the slope of the plane, S_o, the effects of the length, L_o, the normal flow depth, h_o, and the Froude number, F_o.

$$K = \frac{S_o L_0}{h_0 {F_0}^2} \tag{2.6}$$

where the variables with the subscript "o" are the values at the downstream end. Woolhiser and Liggett (1967) showed that the kinematic wave approximation is not appropriate for K smaller than 10 but is good for $K>20$ and $F_0>0.5$ based on numerical experimentation on rising hydrographs at the downstream end of a plane. If the flow near the downstream boundary is subcritical, a numerical problem may arise near the boundary due to the backwater effect. Singh and Aravamuthan (1996) investigated errors in hydrodynamic models for one-dimensional steady state overland flow. They found that the percentage error in water depth over dimensionless distance of the kinematic wave model varied from 6% ($K=30, F_0=1.0$) to 100% ($K=3, F_0=0.1$). The error increased near the upstream end and gradually decreased toward the downstream end. The error was large for small K, but it was lower than 10% at the downstream end with large K, and became negligible when $K=\infty$ regardless of the value of the Froude number, and was relatively small in the diffusion wave model. For example, the error in the diffusion wave model ranged from 0.39% ($K=30, Fr_0=1.0$) to 9% ($K=3, F_0=0.1$) for variable conditions. It should be noted that most of the error occurred near the upstream end. They concluded that the error of the diffusion wave model was considerably lower than the kinematic wave model for low KF_0^2; therefore, the diffusion wave model was preferred over the kinematic wave model for small values of KF_0^2.

Vieira (1983) compared the solutions of the Saint-Venant equations with those of the kinematic, diffusion, and gravity wave approximation, for a range of constant Froude and kinematic wave numbers for one-dimensional shallow surface water flow. Two different types of downstream boundary conditions were used: (1) critical flow; and (2) zero depth gradient condition.

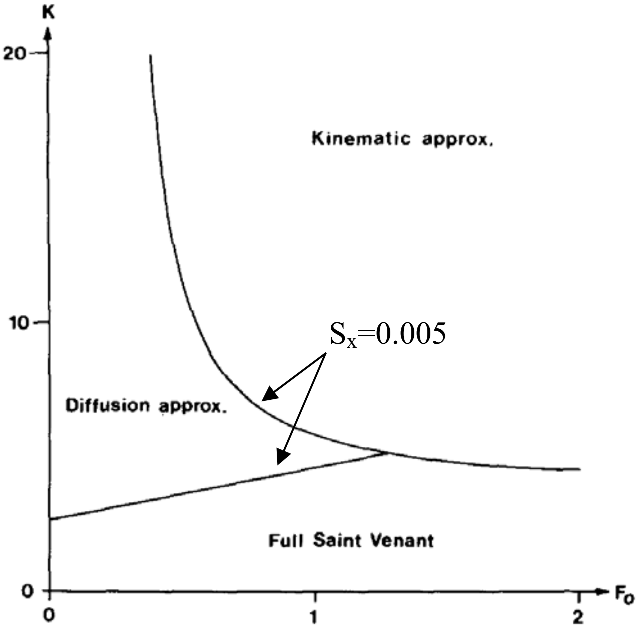


Figure 2.3: Partition of the *K, F_0* field into three zones for zero-depth-gradient downstream boundary conditions. (Vieira, 1983)

By comparing the solutions of hydrodynamic models made for more than 150 *K*-F_0 pairs, Daluz Vieira was able to draw a contour map representing the values of surface slope, S_x (F_0 ,*K*) in the F_0 - *K* field. The contour line corresponding to S_x=0.005 is shown in Figure 2.3. In case of smooth urban slopes where the value of *K* generally ranges from 5 to 20, the kinematic or diffusion wave model may be used depending on the value of F_o where the critical value of F_o is about 0.5 in the figure. The full dynamic wave model must be used for low values of K (<5).

Ponce (1990) developed a linearized diffusion wave model for different cases, which include full inertial, local inertial, convective inertial, and non-inertial. In his numerical experiments, the non-inertial model showed good approximation to the full inertial model for low Froude number flows. Therefore he concluded that the non-inertial diffusion wave model was a better approximation to the full inertial model than either local or convective inertial models.

The theories for the kinematic wave model reviewed in this chapter are mostly for one-dimensional problems. However, the knowledge gained from these kinematic wave theories is crucial for developing a proper multi-dimensional sheet flow model. Even though the kinematic wave model is widely used by many researchers on various topics (Cristina and Sansalone (2003), Akan et. al. (2000), Tisdale et. al. (1999), Tsai and Yang (2005)), it must be noted that most hydrologic processes occur in two-dimensional space and with a temporal variation. Therefore, over-simplification in a physically based model may result in failing in representation of hydrologic or geometric parameters which should be evaluated in a required spatial and temporal resolution (Grayson et. al. (1992)). Therefore, the kinematic wave model may not be applicable to urban flood analyses including sheet flow on impervious pavement surfaces. Backwater effects, ponding, flow over adverse slope, and other flow effects associated with flood control systems must be considered in modeling sheet flow on urban pavement surfaces. The full dynamic wave model would be the best option when it comes to the accuracy of the solution. However, the numerical difficulties and computation cost compromise the benefits of the model, and the necessary condition for using the full dynamic wave model ($K<5$) rarely occurs in rivers and overland flows. Therefore, researchers (Feng and Molz (1997), Jain and Singh (2005), Hromadka and Yen (1986), Lal (1998)) showed that the diffusion wave model is the best model to apply for this purpose. It not only inherits the advantages of the kinematic wave model but also works well for most realistic hydrologic processes.

There have been a few research efforts on modeling stormwater runoff from highway surfaces. Cristina and Sansalone (2003) developed a 1D kinematic wave model of urban pavement rainfall-runoff subject to traffic loadings. They compared the time of concentrations predicted by a kinematic wave model with field measured data. The difference is empirically modeled to an adjusted runoff coefficient. Even though their research is valuable in predicting the amount of stormwater for designing drainage inlets, the model is not able to predict the spatial variation of the stormwater spread within the domain. Anderson et al. (1998) conducted a comprehensive research for improved surface drainage of pavements. The topics encompass various experimental studies and computer modeling efforts. However, they did not develop a proper numerical model for simulating sheet flow on road surfaces due to difficulties in computational cost, convergence, and computational errors. Instead, they built a simple 1D kinematic wave model.

2.3 SUMMARY

In this research, hydraulics of sheet flow near zero cross slope in superelevation transitions is of concern. Even though there are difficulties recognized by fore-mentioned researchers in developing a 2D hydrodynamic diffusion wave model, this model is of choice for this research not only because the hydraulics on complex geometry is of concern but also because the spatial variation of sheet flow needs to be estimated with high accuracy. The main variables of concern contributing to the sheet flow in superelevation transitions are base longitudinal slope, number of lanes, surface roughness, and rainfall intensity. The characteristics of sheet flow on superelevation transitions will be investigated through a numerical experiment by examining the impact of the main variables.

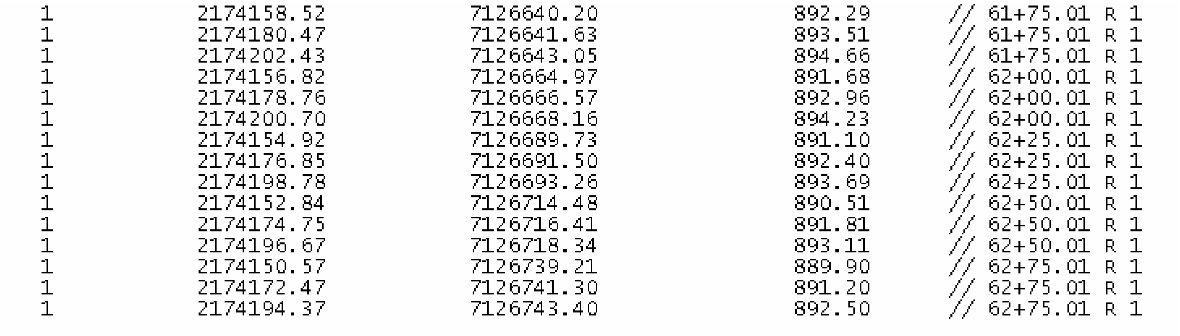
Chapter 3: Grid Generation

3.1 INTRODUCTION

Simulation of sheet flow is sensitive to the domain geometry; therefore it is important to precisely represent the pavement surface near superelevation transition for the numerical model. An interface model is developed for the model to read geometric information in a Digital Terrain Model (DTM) and to create a computational grid space. The mathematical algorithms described in this chapter are presented to create a structured curvilinear grid space based on the DTM. Grid points are defined through a parametric mapping of domain surface for both linear and curved roadway sections.

3.2 GEOMETRY DATA FROM GEOPAK

Bentley's GEOPAK® Civil Engineering Suite is a modular software package for roadway design that is widely used by Departments of Transportation in the United States. GEOPAK calculates superelevation transition for any chain stored in the coordinate geometry database. Since GEOPAK uses coordinate geometry tools to calculate and store design elements, it can provide precise information on the geometry for a complex roadway surface at stations. For the roadway surface profile, one can generate a digital terrain model (DTM) data file (*.dat) in which the 3 dimensional (3D) geometric profile of a roadway is described. The DTM data file is in ASCII format so it may be readily used by other program modules.



```
1          2174158.52          7126640.20           892.29     // 61+75.01 R 1
1          2174180.47          7126641.63           893.51     // 61+75.01 R 1
1          2174202.43          7126643.05           894.66     // 61+75.01 R 1
1          2174156.82          7126664.97           891.68     // 62+00.01 R 1
1          2174178.76          7126666.57           892.96     // 62+00.01 R 1
1          2174200.70          7126668.16           894.23     // 62+00.01 R 1
1          2174154.92          7126689.73           891.10     // 62+25.01 R 1
1          2174176.85          7126691.50           892.40     // 62+25.01 R 1
1          2174198.78          7126693.26           893.69     // 62+25.01 R 1
1          2174152.84          7126714.48           890.51     // 62+50.01 R 1
1          2174174.75          7126716.41           891.81     // 62+50.01 R 1
1          2174196.67          7126718.34           893.11     // 62+50.01 R 1
1          2174150.57          7126739.21           889.90     // 62+75.01 R 1
1          2174172.47          7126741.30           891.20     // 62+75.01 R 1
1          2174194.37          7126743.40           892.50     // 62+75.01 R 1
```

Figure 3.1: An example of geometric information of a roadway in the DTM

A GEOPAK DTM has the information of x-, y-, and z-coordinates of a roadway surface at the center and the end of traffic lanes with a constant interval along the roadway. Figure 3.1 shows the format from a DTM data file. The second column represents the x-coordinate; the third column the y-coordinate; the fourth column the surface elevation; and the last column represents the station number. Three rows are allotted to one station as shown in the last column. At each station, the first row represents the left end point; the second row the center point; and the third row the right end points. One may notice that the DTM in Figure 3.1 is for a roadway with one traffic lane in each direction so there are only 3 points at a station (center, left, and right ends). As the number of traffic lanes increases or decreases, the number of rows for each station also increases or decreases, respectively.

3.3 CURVATURE GEOMETRY

The interface model stores the coordinates of center and side ends of a roadway section. From here we can take a sequence of points $\vec{x}_i = (x_i, y_i)$ that specify the roadway centerline which has curvilinear length ξ_c, as shown in Figure 3.2.

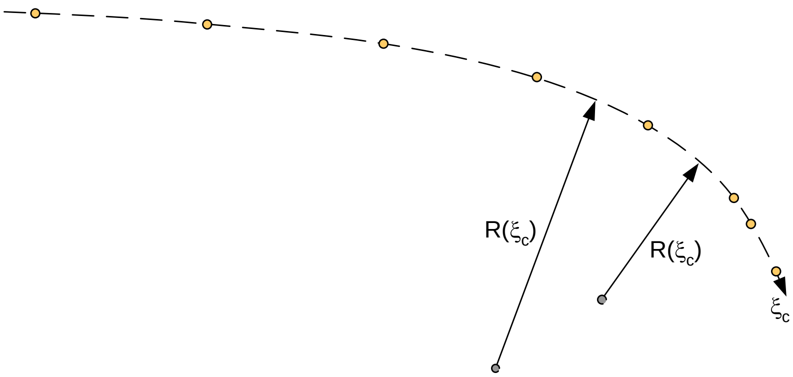


Figure 3.2: Sequence of centerline points with radius of curvature R(ξ_c)

The radius of curvature and center of curvature for each point can be identified based on locations of neighboring points. The geometry for the computational algorithm is shown in Figure 3.3.

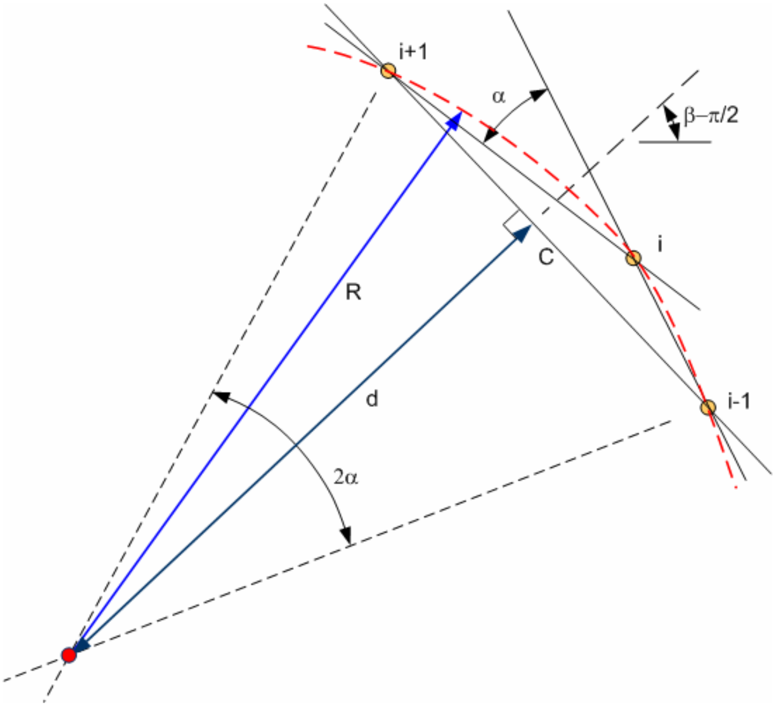


Figure 3.3: Geometry for curvature algorithm

The algorithm for finding the roadway centerline, radius of curvature, and center of curvature proceeds as follows. The change in direction through neighboring points may be found using

$$\alpha_i = \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right) - \tan^{-1}\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) \tag{3.1}$$

The direction of the curvature may be determined from the sign of the α values. A positive α_ι means that the roadway is curved 'counter-clockwise' and a negative value denotes 'clockwise.' The total angle of the sector between rays through points (i-1) and (i+1) is equal to $2\alpha_i$. The length of the chord between these points is

$$C_i = \sqrt{(x_{i+1} - x_{i-1})^2 + (y_{i+1} - y_{i-1})^2} \tag{3.2}$$

With the central angle and chord length, the radius of curvature is

$$R_i = \frac{C_i}{2|\sin(\alpha_i)|} \tag{3.3}$$

The center of the chord, angle of the chord segment, and distance from the center of the chord to the center of curvature are calculated using the following:

$$(x_c, y_c)_i = \left(\frac{x_{i+1} + x_{i-1}}{2}, \frac{y_{i+1} + y_{i-1}}{2}\right) \tag{3.4}$$

$$\beta_i = \tan^{-1}\left(\frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}\right) \tag{3.5}$$

$$d_i = R_i \left|\cos(\alpha_i)\right| \tag{3.6}$$

Since the angle β_i defines the direction of the curvature by taking the inverse tangent of two points, it needs to be modified in the following case:

$$\beta_i = \beta_i + \pi , \qquad if \quad x_{i-1} > x_{i+1} \tag{3.7}$$

d_i in Equation 3.6 is a scalar, so it takes an absolute value of a cosine. With these values the center of curvature is

$$(x_{cc}, y_{cc})_i = (x_c, y_c)_i - (d_i \cos(\gamma_i), d_i \sin(\gamma_i)) \tag{3.8}$$

The position of the center of curvature depends on the direction to which the curve rotates.

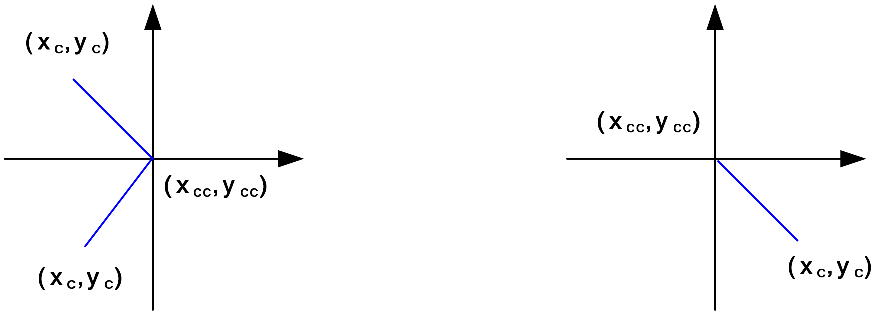
$$\begin{aligned} \gamma_i &= \beta_i - \pi/2 \qquad for \quad \alpha_i < 0 \\ \gamma_i &= \beta_i + \pi/2 \qquad for \quad \alpha_i > 0 \end{aligned} \tag{3.9}$$

3.4 GRID GENERATION FOR CURVATURE

A DTM data file may have an interval of 3 meters or 30 meters between stations, depending on how it is specified. Therefore an objective of grid generation is to refine the coarse DTM grids into reasonably spaced grids for numerical simulation. The algorithm developed up to this point can read the DTM and compute geometric variables that define the centerline curvature. Given the center line geometry through a series of points $(x_c,y_c)_i$ along the roadway centerline with a corresponding series of center of curvature points $(x_{cc},y_{cc})_i$, radius of curvature R_i, and angles θ_i, one may define the locations of (N-1) points along the centerline through curvilinear interpolation using equal “increments” as shown in Figure 3.5. This implies that the distance between points is larger in regions with larger radius of curvature. The direction of the radius of curvature is important because it is later used to determine the placement of interpolated grid points

$$\theta_i = \tan^{-1}\left(\frac{y_c - y_{cc}}{x_c - x_{cc}}\right)_i \tag{3.10}$$

The module defines θ_i to range from zero to 2π so that it can represent the direction of the roadway curvature. The value calculated by Equation 3.10 is modified accordingly (adition of π or 2π) depending on the relative locations of the roadway centerline and center of curvature.



$\theta_i = \theta_i + \pi$ $\qquad\qquad$ $\theta_i = \theta_i + 2\pi$

Figure 3.4: Cases where θ_i needs to be modified

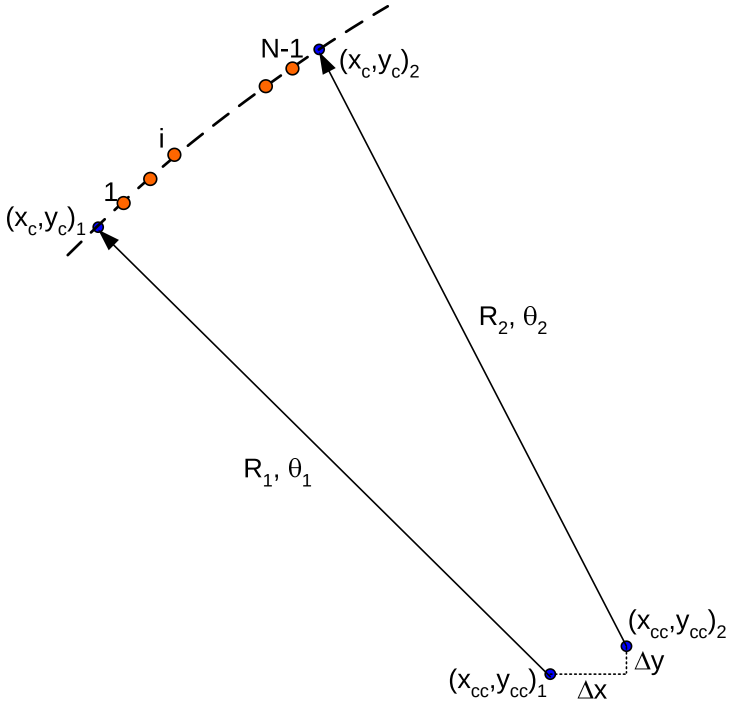


Figure 3.5: Geometry for grid point generation algorithm

In Figure 3.5, the increment of x in the Cartesian coordinate is computed by

$$\Delta x = x_{cc2} - x_{cc1} \tag{3.11}$$

The increments Δy, ΔR, and $\Delta \theta$ are computed similarly. Then, the location of i^{th} grid point in Figure 3.5 may be determined using the following

$$\begin{aligned} x_c(\xi) &= (x_{cc1} + \xi\,\Delta x) + (R_1 + \xi\,\Delta R)\cos(\theta_1 + \xi\,\Delta\theta) \\ y_c(\xi) &= (y_{cc1} + \xi\,\Delta y) + (R_1 + \xi\,\Delta R)\sin(\theta_1 + \xi\,\Delta\theta) \end{aligned} \tag{3.12}$$

In Equation 3.12 the variable ξ ranges from 0 to 1, representing the fractional increment (*i/N)* between endpoints along the curve. This algorithm can be generalized to a curved section of roadway (prototype data space). If W is the roadway width, then a parameter η can be introduced to parameterize the entire roadway as follows

$$
\begin{aligned}
x(\xi,\eta) &= \left(x_{cc1} + \xi\,\Delta x\right) + \left(R_1 + \xi\,\Delta R + (\eta - 0.5)W\right)\cos\left(\theta_1 + \xi\,\Delta\theta\right) \\
y(\xi,\eta) &= \left(y_{cc1} + \xi\,\Delta y\right) + \left(R_1 + \xi\,\Delta R + (\eta - 0.5)W\right)\sin\left(\theta_1 + \xi\,\Delta\theta\right)
\end{aligned}
\tag{3.13}
$$

Equation 3.13 is a parametric mapping of a section of the roadway onto a unit square. In these equations the parameters range $0 < \xi < 1$; $0 < \eta < 1$. The roadway centerline corresponds to $\eta = 0.5$, while the outer curb and inner curb correspond to $\eta = 1$ and $\eta = 0$, respectively. The resulting grid in prototype space is shown schematically in Figure 3.6.

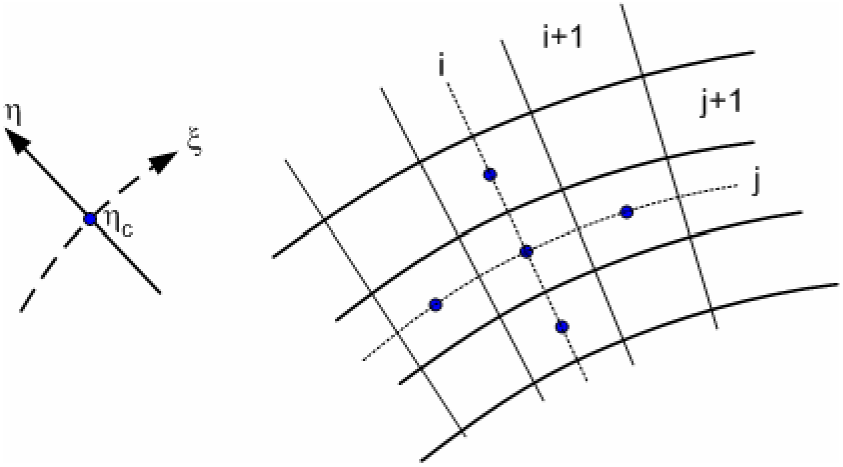


Figure 3.6: Model grid in prototype data space

3.5 CHARACTERIZATION OF THE PARAMETRIC MAPPING

The transformation between the Cartesian coordinate (x,y) and a curvilinear coordinate system (ξ,η) requires partial derivatives of the coordinate transformation functions. Differentiating Equation 3.13 with respect to ξ and η, one gets

$$
\begin{aligned}
\frac{\partial x}{\partial \xi} &= \Delta x + \Delta R \cos(\theta_1 + \xi\,\Delta\theta) - \Delta\theta\left(R_1 + \xi\,\Delta R + (\eta - 0.5)W\right)\sin(\theta_1 + \xi\,\Delta\theta) \\
\frac{\partial y}{\partial \xi} &= \Delta y + \Delta R \sin(\theta_1 + \xi\,\Delta\theta) + \Delta\theta\left(R_1 + \xi\,\Delta R + (\eta - 0.5)W\right)\cos(\theta_1 + \xi\,\Delta\theta) \\
\frac{\partial x}{\partial \eta} &= W\cos(\theta_1 + \xi\,\Delta\theta) \\
\frac{\partial y}{\partial \eta} &= W\sin(\theta_1 + \xi\,\Delta\theta)
\end{aligned}
\tag{3.14}
$$

The parametric representation of the coordinate system is useful since length and area transformations can be calculated. The length scaling factors for segments associated with changes in ξ and η are

$$
\begin{aligned}
L_\xi(\xi,\eta) &= \sqrt{\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2} \\
L_\eta(\xi,\eta) &= \sqrt{\left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2} = W
\end{aligned}
\tag{3.15}
$$

The total length of a segment can be found by integrating the length scaling.

$$\xi - Length = \int_0^1 L_\xi(\xi,\eta)\, d\xi \tag{3.16a}$$

$$\eta - Length = \int_0^1 L_\eta(\xi,\eta)\, d\eta \tag{3.16b}$$

Note that Equation 3.16(a) is for the interval between stations at η position and Equation 3.16(b) is the same as the total width of the roadway. The size of a grid cell can be computed similarly.

$$\ell(\xi,\eta) = L_\xi(\xi,\eta)\Delta\xi \tag{3.17a}$$

$$w(\xi,\eta) = W\,\Delta\eta \tag{3.17b}$$

In Equation 3.17 $\Delta\xi=1/N_\xi$, $\Delta\eta=1/N_\eta$ where N_ξ and N_η are the number of grid cells within the sector in each local coordinate direction. The area increments transform according to the Jacobian defined as

$$J(\xi,\eta) = \begin{vmatrix} \partial x/\partial\xi & \partial y/\partial\xi \\ \partial x/\partial\eta & \partial y/\partial\eta \end{vmatrix} = \left(\frac{\partial x}{\partial\xi}\right)\left(\frac{\partial y}{\partial\eta}\right) - \left(\frac{\partial y}{\partial\xi}\right)\left(\frac{\partial x}{\partial\eta}\right) \tag{3.18}$$

Within the Cartesian coordinate system the unit vectors along the ξ and η curves are calculated as follows:

$$\begin{aligned} \hat{u}_\xi &= \frac{1}{L_\xi}\frac{\partial x}{\partial\xi}\hat{i} + \frac{1}{L_\xi}\frac{\partial y}{\partial\xi}\hat{j} \\ \hat{u}_\eta &= \frac{1}{L_\eta}\frac{\partial x}{\partial\eta}\hat{i} + \frac{1}{L_\eta}\frac{\partial y}{\partial\eta}\hat{j} \end{aligned} \tag{3.19}$$

The inner product of these unit vectors represents the cosine of the angle between these unit vectors.

$$\cos(\omega) = \hat{u}_\xi \cdot \hat{u}_\eta \tag{3.20}$$

With Equation 3.19, Equation 3.20 takes an algebraic form.

$$\cos(\omega) = \frac{1}{L_\xi}\left(\Delta x\cos(\theta_1 + \xi\,\Delta\theta) + \Delta y\sin(\theta_1 + \xi\,\Delta\theta) + \Delta R\right) \tag{3.21}$$

It is only for the special case $\cos(\omega) = 0$ that the parametric curves are orthogonal. Figure 3.7 shows an example of a computed curvilinear grid based on Equations 3.13 to 3.21. In this figure, the solid circles correspond to the roadway centerline. The solid diamonds correspond to the refined grid points on the domain. Between the stations *i-1* and *i,* the spacing in the ξ-direction is visually non-uniform, as expected because of the rapidly increasing radius of curvature.

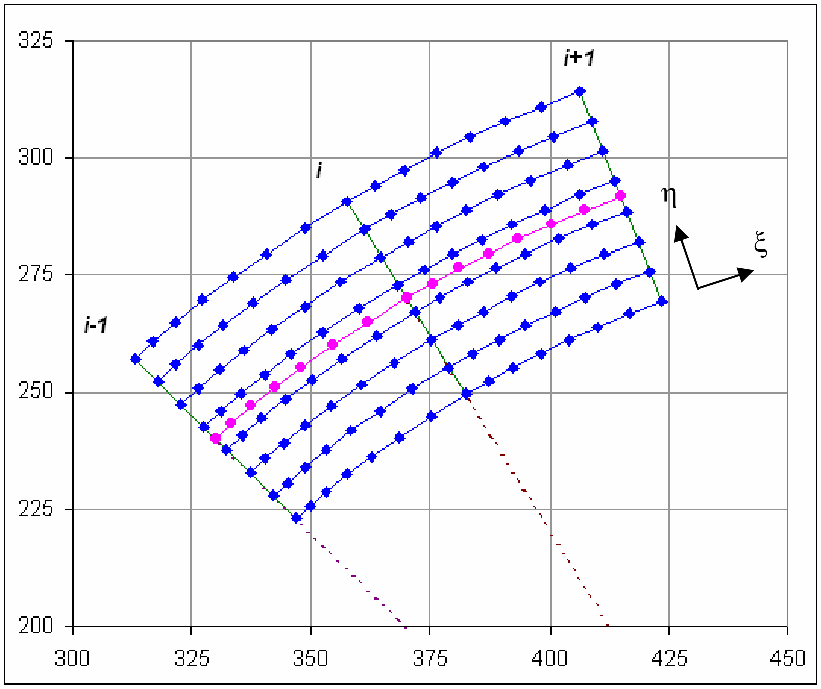


Figure 3.7: Grid layout for a domain with a curved roadway

In order to evaluate the sensitivity of the transformation, a 6 m long and 7.3 m wide curved sector with the design speed of 96 km/hr is considered. At the center of the sector ($\xi = 0.5$), the radius of curvature is $R = 365$m. This sector has $\Delta\theta = 0.01667$ radian. The radii of the outer and inner curb are $R_o = 369.4$ m and $R_i = 362.1$ m. Using standard cylindrical coordinate formula, the area of the section is $A = 0.5\Delta\theta\ (R_o^2 -$

R_i^2) = 44.602 m^2. Evaluating the Jacobian at the center (treating the entire sector as one grid cell) gives A = J(0.5,0.5) = 44.5935 m^2. Using a 5x5 division of the unit area gives A = (1/5) (J(1/10, 1/10) + J(1/10, 3/10) + … + J(9/10, 9/10)) = 44.5935 m^2. Clearly the area transformation is well behaved and the section area is approximately A = 44.5935 m^2. Application of the Jacobian to any sub-area will give results of equal or increased accuracy.

3.6 ALGORITHM FOR LINEAR SECTION

For the case of a linear segment within the domain, mathematical equations should be developed similar to a curved sector. The geometry of a straight roadway is simple, as shown in Figure 3.8. In this case, the roadway can be considered as a rectangular domain rotated by an angle δ. Therefore a rectangular grid may be easily defined by the center line geometry and the direction of the roadway.

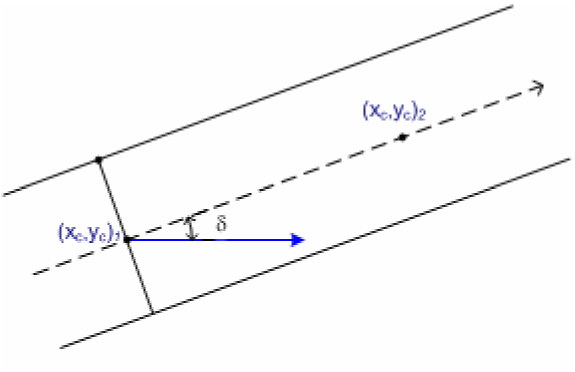


Figure 3.8: Geometry for grid generation for straight section of roadway

The direction of the roadway is defined as the direction of the roadway with respect to the horizontal axis.

$$\delta = \tan^{-1}\left(\frac{y_{c2} - y_{c1}}{x_{c2} - x_{c1}}\right) \tag{3.22}$$

The direction angle (δ) should be modified based on the direction of the roadway as shown in Figure 3.4 such that

$$\begin{aligned} \delta &= \delta + \pi && \text{if} \quad x_{c1} < x_{c2} \\ \delta &= \delta + 2\pi && \text{if} \quad \delta < 0 \end{aligned} \tag{3.23}$$

Now the grid points on a straight roadway can be linearly interpolated.

$$\text{for} \quad \alpha_i < 0 \qquad \begin{aligned} x(\xi,\eta) &= x_{c1} + \xi L_{12}\cos(\delta) - (\eta - 0.5)W\sin(\delta) \\ y(\xi,\eta) &= y_{c1} + \xi L_{12}\sin(\delta) + (\eta - 0.5)W\cos(\delta) \end{aligned} \tag{3.24a}$$

$$\text{for} \quad \alpha_i > 0 \qquad \begin{aligned} x(\xi,\eta) &= x_{c1} + \xi L_{12}\cos(\delta) + (\eta - 0.5)W\sin(\delta) \\ y(\xi,\eta) &= y_{c1} + \xi L_{12}\sin(\delta) - (\eta - 0.5)W\cos(\delta) \end{aligned} \tag{3.24b}$$

In these equations, $(x,y)_{c1}$ are the center line coordinates in the DTM, L_{12} is the distance between stations, and the parameters range $0<\xi<1$, $0<\eta<1$. Since the module uses three points to compute geometric values for the curvature, the first and the last points of a curve can not explicitly have the geometric information. These points take the values of neighboring points inside the curvature so the interface connects a straight section to a curvature smoothly. For transformation between (x,y) and (ξ,η) representations, take the partial derivative for Equations 3.23 and 3.24.

$$\frac{\partial x}{\partial \xi} = L_{12}\cos(\theta), \quad \frac{\partial y}{\partial \xi} = L_{12}\sin(\theta)$$

$$\frac{\partial x}{\partial \eta} = \mp W\sin(\theta), \quad \frac{\partial y}{\partial \eta} = \pm W\cos(\theta) \tag{3.25}$$

From the transformation derivatives (Equation 3.15) it is clear that $L_{\xi} = L_{12}$ and $L_{\eta} = W$. The Jacobian of transformation (Equation 3.18) for a linear section is $J(\xi,\eta) = L_{12}W$. This result shows that there is no spatial distortion in the transformation of linear sections.

3.7 GEOMETRY DATA SCREENING

The numerical model recognizes the roadway shape with the sign of the angle shown in Equation 3.1. If α_ι is zero, the shape of the roadway at the station *i* is linear; if α_ι is non-zero then it is curved. Whereas the precision of a DTM can be controlled manually up to 4 digits after the decimal, it turned out that the accuracy of the data is not enough to use raw values. The data points in a DTM have significant "noise" that may cause the interface model to fail to read the geometric information from the DTM correctly. For instance, a line-curve-line shaped domain may have α_ι's at stations as shown in Figure 3.10. In this figure, the hollow circles represent the alpha values of the roadway center curve with original geometry data, and the solid squares are filtered values. The existence of noise is apparent in the data and causes error in determining the roadway direction and resulting grid layout. A filtering technique that decreases data-noise effects is developed: alpha values are computed for the roadway sides as well as at the center of the domain and the average of these values rounded up at 0.001 represents the direction of the roadway at the station. The results from such filtering are shown with blue squares in Figure 3.9. The filtered data is reasonable to represent the original shape of the domain; α_ιs are zero for the linear sections and are constant with a non-zero value (0.012) throughout the curved section with two exceptional points at transitions.

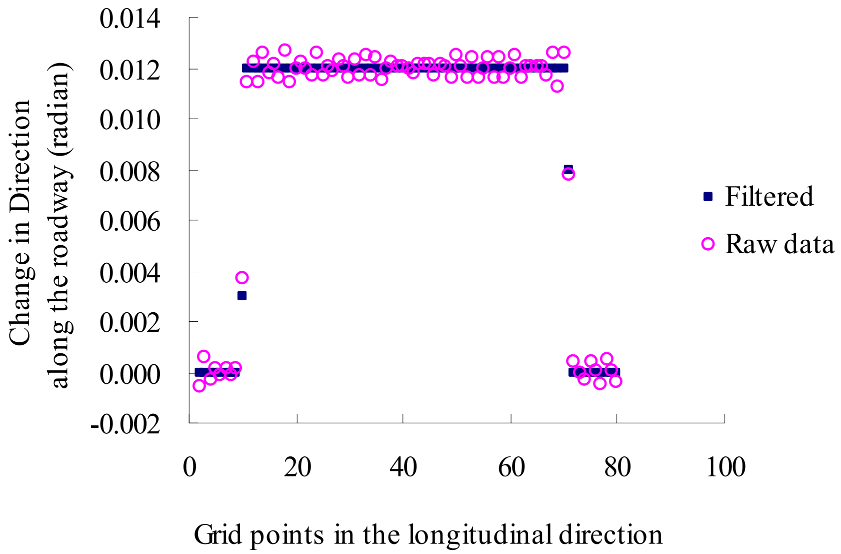
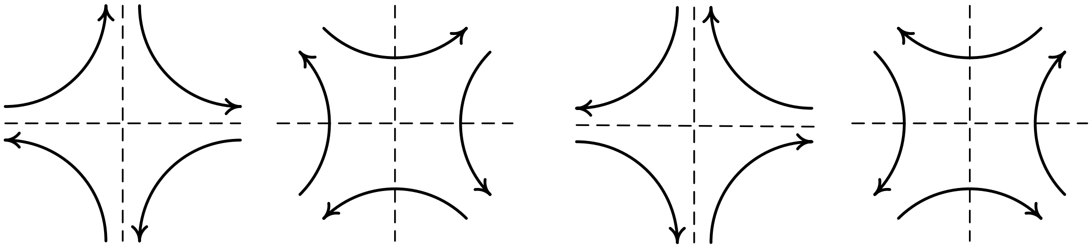


Figure 3.9: Use of filtering to minimize "data-noise" in roadway direction

Grid interval can be defined differently in the ξ− and η−directions. A grid with large $\Delta\xi$ and small $\Delta\eta$ for a long and narrow curved roadway section can save computation costs without undermining the reliability of the result. The algorithm defines grid points at the center of grid cells. Surface elevation is binomially interpolated from the DTM at the corners of cells then the values at corners of a cell are averaged to get the elevation at the center.

3.8 RESTRICTIONS ON THE GEOMETRY OF ROADWAY

It is restricted in the model that a curved domain should be either clockwise or counter-clockwise. The roadway may also be straight throughout the domain. An S-shaped domain which may have both the ‘clockwise’ and ‘counter-clockwise’ directions cannot be modeled with the present formulation. This might be trivial because S-curves are normally designed to have a straight section in the middle. With this restriction, the number of typical patterns of curvatures may be limited to those shown in Figure 3.10. These patterns will be used to test the interface model.



Counter-clockwise

Clockwise

Figure 3.10: Curvature profiles for a roadway section

The restrictions in the applicability of the model on specific types of geometry are stated in the following list. The proposed model is not applicable for simulation if

1. an apex or lowest point exists inside of the domain, not on the boundary. A domain with crest vertical curves or sag vertical curves is not applicable.
2. a domain is horizontally S-shape curve. This is a rare case in practice but if it exists, such geometry is not allowed in the model due to the existence of the inflection point between curves.

3. a domain is a converging or diverging section. The model does not recognize a start or end of traffic lanes within the domain.
4. the road is a divided highway. In this case, the model should be applied to each set of lanes separately. Therefore, the model recognizes the median as one side and the roadside another. If there exists drainage inlets along the median, these may be treated as curb inlets.

Chapter 4: Model Development

4.1 INTRODUCTION

A simplified form of the Saint-Venant equations based on diffusion wave formulation may be used to model sheet and gutter flow on complex pavement surfaces. This chapter outlines the development of a diffusion wave model for sheet flow, development of appropriate boundary conditions, and model testing. The Manning's equation is used to define the hydraulic friction slop. A set of finite volume based discrete equations is solved implicitly by a general conjugate gradient method with incomplete Cholesky decomposition.

4.2 SAINT VENANT EQUATIONS FOR SHEET FLOW

Consider a sheet flow over a plane where no infiltration occurs (Figure 4.1). The depth of the sheet flow is relatively small compared with the width and length of the stream. Therefore it may be reasonable to assume that the vertical component of the sheet flow is negligible. Furthermore, it is assumed that rainfall is uniform in space and vertical in direction.

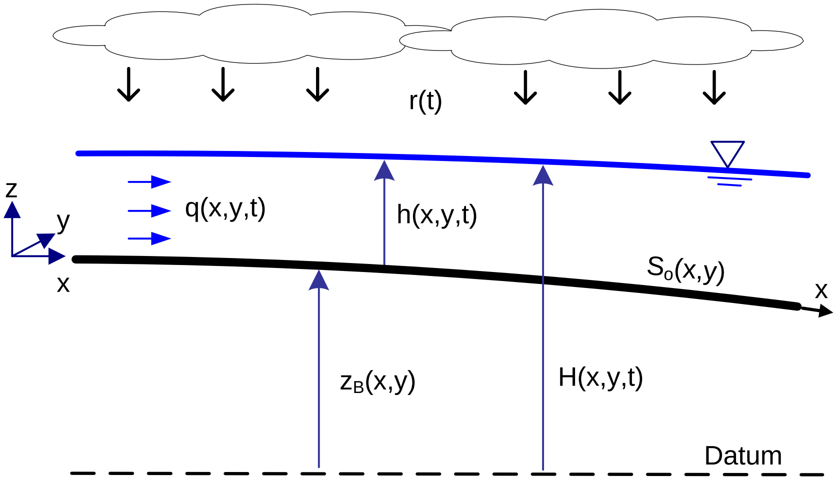


Figure 4.1: Overland flow over a plane

The surface slope, S_o, is defined as positive for a down slope $S_{oi} = -\partial z_B / \partial x_i$ and the total head $H(x,y,t) = h(x,y,t) + z_B(x,y)$ is the main variable of the mathematical model for which a set of nonlinear differential equations is solved. The general constitutive equations for 2D flow include the continuity equation,

$$\frac{\partial H}{\partial t} + \frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} - r = 0 \tag{4.1}$$

and two full momentum equations.

$$\begin{aligned} \frac{\partial q_x}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x^2}{h}\right) + \frac{\partial}{\partial y}\left(\frac{q_x q_y}{h}\right) + gh\left[\frac{\partial h}{\partial x} + S_{fx} - S_{ox}\right] = 0 \\ \frac{\partial q_y}{\partial t} + \frac{\partial}{\partial y}\left(\frac{q_y^2}{h}\right) + \frac{\partial}{\partial x}\left(\frac{q_x q_y}{h}\right) + gh\left[\frac{\partial h}{\partial y} + S_{fy} - S_{oy}\right] = 0 \end{aligned} \tag{4.2}$$

To simplify notation, the inertial terms may be written in the following form:

$$A_x = \frac{1}{gh}\left(\frac{\partial q_x}{\partial t} + \frac{\partial}{\partial x}\left(\frac{q_x^2}{h}\right) + \frac{\partial}{\partial y}\left(\frac{q_x q_y}{h}\right)\right) \tag{4.3}$$

$$A_y = \frac{1}{gh}\left(\frac{\partial q_y}{\partial t} + \frac{\partial}{\partial y}\left(\frac{q_y{}^2}{h}\right) + \frac{\partial}{\partial x}\left(\frac{q_x q_y}{h}\right)\right)$$

Then the momentum equations may be interpreted such that the friction slope is the summation of the bed slope, depth gradient, and the inertial term.

$$\begin{aligned} S_{fx} - S_{ox} + \frac{\partial h}{\partial x} + A_x &= 0 \\ S_{fy} - S_{oy} + \frac{\partial h}{\partial y} + A_y &= 0 \end{aligned} \tag{4.4}$$

4.3 DIFFUSION WAVE MODEL WITH MANNING'S EQUATION

Highway drainage is usually modeled using a combination of Manning's equation and kinematic wave theory for estimation of the time of concentration (Brown et al., 2001). However, because of the nonuniform drainage flow paths through superelevation transitions, diffusion wave models are employed, allowing for the effects of lateral pressure gradients to be included.

$$\begin{aligned} S_{fx} - S_{ox} + \frac{\partial h}{\partial x} = 0, \text{ or } S_{fx} = -\frac{\partial H}{\partial x} \\ S_{fy} - S_{oy} + \frac{\partial h}{\partial y} = 0, \text{ or } S_{fy} = -\frac{\partial H}{\partial y} \end{aligned} \tag{4.5}$$

where $H=h+z_B$. In a vector form, these are simply expressed such that

$$\vec{S}_f = -\nabla H \tag{4.6}$$

If the depth gradient is dropped out from Equation 4.5 then the friction slope becomes the same as surface slope. This simplest form of the Saint Venant equations is called the kinematic wave model.

$$\vec{S}_f = \vec{S}_o \tag{4.7}$$

Manning's equation may be expressed in a vector form such that

$$\vec{q} = \frac{h^{5/3}}{n\sqrt{S_f}} \vec{S}_f \tag{4.8}$$

where h = flow depth, n = Manning coefficient, and S_f = magnitude of friction slope. Substituting Equation 4.6 into Equation 4.8 and rearranging such that scalar values are represented by a coefficient gives

$$\vec{q} = -D_h \nabla H \tag{4.9}$$

where D(h) is a nonlinear diffusion coefficient, a function of water depth, Manning's coefficient, and surface friction.

$$D_h = \frac{h^{5/3}}{n\sqrt{S_f}} \tag{4.10}$$

In this equation, S_f in the denominator is the magnitude of the total friction slope vector. The diffusion coefficient is a scalar value representing the diffusivity of the flow for given conditions. If the free surface of water is locally flat ($S_f = 0$, from Equation 4.6), D_h becomes infinity. Therefore, a limiting condition is necessary to eliminate the singularity introduced by D_h in the algorithm: If the friction slope is locally zero, then the algorithm forces the diffusion coefficient to be zero. Physically, this implies water does not flow if free surface gradient of sheet flow is zero.

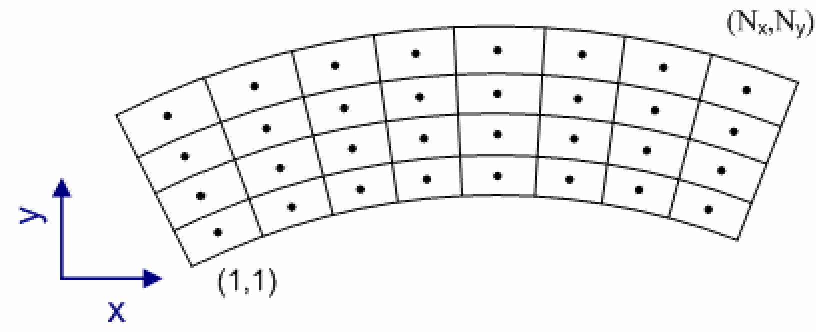
Flow rate is now expressed as a function of water depth. Therefore, Equation 4.9 can be substituted into the continuity equation (Eq. 4.1) to get the full expression of a diffusion wave model: a 2D second order nonlinear parabolic partial differential equation (PDE).

$$\frac{\partial H}{\partial t}=\frac{\partial}{\partial x}\left(D(h)\frac{\partial H}{\partial x}\right)+\frac{\partial}{\partial y}\left(D(h)\frac{\partial H}{\partial y}\right)+r \tag{4.11}$$

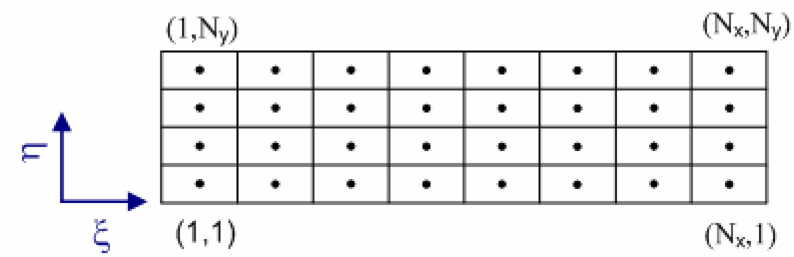
In formulating the numerical model using the finite volume method, Equation 4.11 is not used directly. Instead, each grid cell is considered as a control volume, and the continuity equation is applied for each cell. The resulting system of equations is combined to yield a domain equation to solve for either the flow depth *h* or water surface elevation *H* as a function of time

4.4 NUMERICAL MODEL DEVELOPMENT

Equation 4.11 has no analytical solution due to the nonlinear diffusion coefficient; therefore only an approximate solution is available through numerical computation. For the numerical computation, all the geometric properties of the curvilinear grids (physical space) developed in Chapter 3 are transformed into rectangular grids (computational space) as shown in Figure 4.2. In this figure, ξ represents the longitudinal direction of roadway and η, the transverse direction. Indices *i* and *j* range from $(1,1)$ to (N_x,N_y). The curvilinear shape of the physical domain is straightened up making the shape rectangular in the computational domain. Geometric properties are conserved through the transformation properties.



(a) Curvilinear grids in physical space (x,y)



(b) Rectangular grids in computational space (ξ,η)

Figure 4.2: Transformation of grid space

The finite volume method (FVM) is used to develop the numerical model equations since it more easily provides local mass balance on nonuniform grids compared with other finite-grid schemes. For space discretization, 3-point central differencing is used and the Crank-Nicolson method is employed for time differencing. The equation can be formulated for the computational grid cell shown in Figure 4.3.

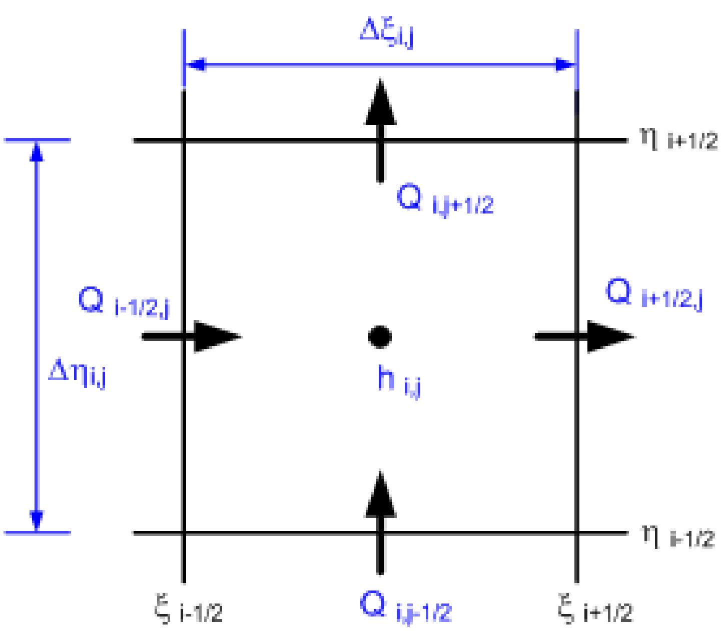


Figure 4.3: Transformed grid for cell (i,j)

When applied to the i,j-grid cell shown in Figure 4.3 the continuity equation (Equation 4.1) is written

$$\frac{dH_{i,j}}{dt}+\frac{1}{A_{i,j}}\left(Q_{i,j-1/2}+Q_{i-1/2,j}+Q_{i,j+1/2}+Q_{i+1/2,j}\right)=r \tag{4.12}$$

In Equation 4.12, $h_{i,j}$ is the depth in the cell [the total head $H_{i,j}$ could equally be used as the primary variable], $A_{i,j}$ is the grid cell area, the Q's are the volume discharges outward across the grid cell boundary, and r is the constant rainfall rate. Within the transformed coordinate system, the main variable, total head (H), is evaluated at the center of cells, while the flux (Q) is estimated at the cell boundaries. Each cell has four boundaries interfacing with neighboring cells. The longitudinal fluxes (ξ-axis) associated with the cell (i,j) may be discretized as follows.

$$
\begin{aligned}
Q_{i-1/2,j} &= -\frac{2\,D_{i-1/2,j}}{\Delta\xi_{i-1,j}+\Delta\xi_{i,j}}\Delta\eta_{i,j}\left(H_{i-1,j}-H_{i,j}\right)\\
Q_{i+1/2,j} &= -\frac{2\,D_{i+1/2,j}}{\Delta\xi_{i,j}+\Delta\xi_{i+1,j}}\Delta\eta_{i,j}\left(H_{i+1,j}-H_{i,j}\right)
\end{aligned}
\qquad (4.13a)
$$

Similarly, fluxes in the transverse direction (η-axis) take the following form.

$$
\begin{aligned}
Q_{i,j-1/2} &= -\frac{2\,D_{i,j-1/2}}{\Delta\eta_{i,j-1}+\Delta\eta_{i,j}}\Delta\xi_{i,j}\left(H_{i,j-1}-H_{i,j}\right)\\
Q_{i,j+1/2} &= -\frac{2\,D_{i,j+1/2}}{\Delta\eta_{i,j}+\Delta\eta_{i,j+1}}\Delta\xi_{i,j}\left(H_{i,j+1}-H_{i,j}\right)
\end{aligned}
\qquad (4.13b)
$$

In order to implement the model, Equation 4.12 must be transformed back to the prototype data space. The transformation to the roadway grid system requires transformation of boundary and area sections. To parameterize the cell transformation, one may introduce "cell conveyance" coefficients ($C_{i,j}$) which vary in space and time. First, the boundary discharge terms from Equation 4.13 are divided by the cell area (see Equation 4.12) and the resulting cell conductance terms are defined by

$$
\begin{aligned}
\left(C_\xi\right)_{i-1/2,j} &= \frac{2\,D_{i-1/2,j}}{\Delta\xi_{i,j}\left(\Delta\xi_{i-1,j}+\Delta\xi_{i,j}\right)}\\
\left(C_\xi\right)_{i+1/2,j} &= \frac{2\,D_{i+1/2,j}}{\Delta\xi_{i,j}\left(\Delta\xi_{i,j}+\Delta\xi_{i+1,j}\right)}\\
\left(C_\eta\right)_{i,j-1/2} &= \frac{2\,D_{i,j-1/2}}{\Delta\eta_{i,j}\left(\Delta\eta_{i,j-1}+\Delta\eta_{i,j}\right)}\\
\left(C_\eta\right)_{i,j+1/2} &= \frac{2\,D_{i,j+1/2}}{\Delta\eta_{i,j}\left(\Delta\eta_{i,j}+\Delta\eta_{i,j+1}\right)}
\end{aligned}
\qquad (4.14)
$$

When transformed to the roadway grid using Equation 3.17 these terms become

$$
\left(C_\xi\right)_{i-1/2,j} = \frac{2\,D_{i-1/2,j}}{\ell_{i,j}\left(\ell_{i-1,j}+\ell_{i,j}\right)} \qquad (4.15)
$$

$$(C_\xi)_{i+1/2,j} = \frac{2D_{i+1/2,j}}{\ell_{i,j}(\ell_{i+1,j} + \ell_{i,j})^2}$$

$$(C_\eta)_{i,j-1/2} = \frac{2D_{i,j-1/2}}{w_{i,j}(w_{i,j-1} + w_{i,j})}$$

$$(C_\eta)_{i,j+1/2} = \frac{2D_{i,j+1/2}}{w_{i,j}(w_{i,j} + w_{i,j+1})}$$

In equation 4.15 $\ell_{i,j}$ and $w_{i,j}$ are the length and width of the cell *i,j* as defined in Equation 3.17. The ranges of (i,j) indices are $1 \le i \le N_x$ and $1 \le j \le N_y$ where N_x is the number of grid points in the ξ direction and N_y is that in the η direction over the entire domain. The effective nonlinear diffusion coefficient is calculated using the interpolated values of the flow depth and friction slope at the center of the boundary between the cells. For example, the water depth at the interface between $P_{i,j}$ and $P_{i+1,j}$ is calculated from

$$h_{i+1/2,j} = \frac{h_{i,j}\ell_{i+1,j} + h_{i+1,j}\ell_{i,j}}{\ell_{i+1,j} + \ell_{i,j}} \qquad (4.16)$$

For the ξ-component of the friction slope Equation 4.6 is discretized on the boundary of cells. For instance, the friction slope between the cells *i,j* and *i+1,j* is approximated by

$$(S_{f\xi})_{i+1/2,j} = \frac{2(H_{i,j} - H_{i+1,j})}{\ell_{i,j} + \ell_{i+1,j}} \qquad (4.17)$$

The η-components of the friction slope ($S_{f\eta}$) for the transverse gradient can be calculated using expressions similar to Equation 4.17. The average transverse gradient at the boundary between cell (i,j) and (i+1,j) requires values of $(S_{f\eta})_{i,j+1/2}$, $(S_{f\eta})_{i+1,j+1/2}$, $(S_{f\eta})_{i+1,j-1/2}$, and $(S_{f\eta})_{i,j-1/2}$. The interpolated value of the η-component at the center of the ξ-face of the cell is then calculated using

$$\left(S_{f\eta}\right)_{i+1/2,j} = \frac{\left(\left(S_{f\eta}\right)_{i,j+1/2} + \left(S_{f\eta}\right)_{i,j-1/2}\right)\ell_{i+1,j} + \left(\left(S_{f\eta}\right)_{i+1,j+1/2} + \left(S_{f\eta}\right)_{i+1,j-1/2}\right)\ell_{i,j}}{2\left(\ell_{i+1,j} + \ell_{i,j}\right)} \tag{4.18}$$

The magnitude of the friction slope at the interface is calculated using

$$\left(S_f\right)_{i+1/2,j} = \sqrt{\left(S_{f\xi}\right)^2_{i+1/2,j} + \left(S_{f\eta}\right)^2_{i+1/2,j}} \tag{4.19}$$

With Equations 4.16 through 4.19, the discrete form of the diffusion coefficient on the cell interface takes the form

$$D_{i+1/2,j} = \frac{h^{5/3}_{i+1/2,j}}{n_{i,j}\sqrt{\left(S_f\right)_{i+1/2,j}}} \tag{4.20}$$

With Equations 4.16, 4.19, and 4.20, the conveyance coefficient can be evaluated from Equation 4.15. Combining all of the terms, Equation 4.11 may be written

$$\frac{dh_{i,j}}{dt} + f(h) + f(z) - r = 0 \tag{4.21}$$

In Equation 4.21

$$\begin{aligned} f(h) = {} & \left(\left(C_\xi\right)_{i-1/2,j} + \left(C_\eta\right)_{i,j-1/2} + \left(C_\xi\right)_{i+1/2,j} + \left(C_\eta\right)_{i,j+1/2}\right)h_{i,j} - \left(C_\xi\right)_{i-1/2,j}h_{i-1,j} \\ & - \left(C_\xi\right)_{i+1/2,j}h_{i+1,j} - \left(C_\eta\right)_{i,j-1/2}h_{i,j-1} - \left(C_\eta\right)_{i,j+1/2}h_{i,j+1} \end{aligned}$$

$$\begin{aligned} f(z) = {} & \left(\left(C_\xi\right)_{i-1/2,j} + \left(C_\eta\right)_{i,j-1/2} + \left(C_\xi\right)_{i+1/2,j} + \left(C_\eta\right)_{i,j+1/2}\right)z_{B\,i,j} - \left(C_\xi\right)_{i-1/2,j}z_{B\,i-1,j} \\ & - \left(C_\xi\right)_{i+1/2,j}z_{B\,i+1,j} - \left(C_\eta\right)_{i,j-1/2}z_{B\,i,j-1} - \left(C_\eta\right)_{i,j+1/2}z_{B\,i,j+1} \end{aligned}$$

In practical applications, the water surface elevation (*H*) is usually several digits larger than water depth (*h*) in order-of-magnitude. Therefore, the water depth term is used as the main variable to reduce truncation error which may be caused by the difference between water depth and surface elevation in the order of magnitude ($h/z_B \sim 10^{-6}$). The final system of equations is evaluated using the Crank-Nicolson method (Ferziger and Peric, 2002) which gives

$$\frac{{h_{i,j}}^{k+1} - {h_{i,j}}^{k}}{\Delta t} + \frac{1}{2}\left(f(h) + f(z)\right)^{k+1} + \frac{1}{2}\left(f(h) + f(z)\right)^{k} - r = 0 \tag{4.22}$$

The superscript refers to the time level at which the variable and coefficients are evaluated. The system of equations is solved iteratively using a conjugate gradient method solver with an inner loop for updating the nonlinear coefficient and outer loop for convergence check.

4.5 SOLUTION PROCESS

The transformed grid is shown in Figure 4.4. There are a total of N_R rows and N_C columns in the domain. The model system of equation should be formulated:

$$\underline{\underline{A}}\,\underline{x} = \underline{f} \tag{4.23}$$

The unknown vector *x* has global components of dimension, $N = N_R \cdot N_C$.

$$\begin{aligned} \underline{x} &= \left[h_{1,1} \;\; h_{1,2} \;\ldots\; h_{1,N_R} \;\; h_{2,1} \;\ldots\; h_{N_C,N_R-1} \;\; h_{N_C,N_R}\right]^T \\ x_n &= h_{i,j} \qquad\qquad \text{where} \quad n = i \cdot N_R + j \end{aligned} \tag{4.24}$$

The local unknown $h_{i,j}$ is located as global component x_n. The matrix $\underline{\boldsymbol{A}}$ is a square *N* by *N* matrix with components $a_{m,n}$ where $1 \le m \le N,\ 1 \le n \le N$. The column vector *f* is numbered the same as the unknown vector *x*.

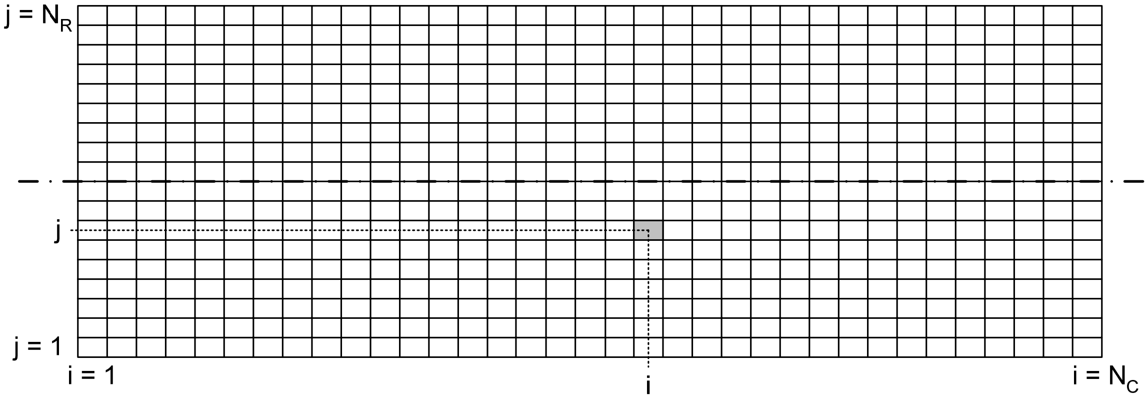


Figure 4.4: Transformed model grid

With the row-column numbering system, Equation 4.23 may be written

$$a_n x_{n-N_R}^{k+1} + b_n x_{n-1}^{k+1} + c_n x_n^{k+1} + d_n x_{n+1}^{k+1} + e_n x_{n+N_R}^{k+1} = f_n \tag{4.25}$$

For the set of equations discretized in 2D space shown in Equation 4.25, the coefficient matrix *A* becomes pentadiagonal which means there are 2 upper diagonals and 2 lower diagonals with main diagonal band. For the index notation defined in Equation 4.24 the components of the matrix *A* are

$$\begin{aligned} a_n &= -\left(C_\xi\right)_{i-1/2,j}^{k+1} \\ b_n &= -\left(C_\eta\right)_{i,j-1/2}^{k+1}, \\ c_n &= \left(C_\xi\right)_{i-1/2,j}^{k+1} + \left(C_\eta\right)_{i,j-1/2}^{k+1} + \left(C_\xi\right)_{i+1/2,j}^{k+1} + \left(C_\eta\right)_{i,j+1/2}^{k+1} + 2/\Delta t \\ d_n &= -\left(C_\eta\right)_{i,j+1/2}^{k+1} \\ e_n &= -\left(C_\xi\right)_{i+1/2,j}^{k+1} \end{aligned} \tag{4.26a}$$

The RHS vector *f* is

$$
\begin{aligned}
f_n = {} & (2/\Delta t)h_{i,j}^{k} + 2r \\
& -(C_\xi)_{i-1/2,j}^{k+1}\left(z_{B\,i,j} - z_{B\,i-1,j}\right) - (C_\xi)_{i+1/2,j}^{k+1}\left(z_{B\,i,j} - z_{B\,i+1,j}\right) \\
& -(C_\eta)_{i,j-1/2}^{k+1}\left(z_{B\,i,j} - z_{B\,i,j-1}\right) - (C_\eta)_{i,j+1/2}^{k+1}\left(z_{B\,i,j} - z_{B\,i,j+1}\right) \\
& -(C_\xi)_{i-1/2,j}^{k}\left(H_{i,j}^{k} - H_{i-1,j}^{k}\right) - (C_\xi)_{i+1/2,j}^{k}\left(H_{i,j}^{k} - H_{i+1,j}^{k}\right) \\
& -(C_\eta)_{i,j-1/2}^{k}\left(H_{i,j}^{k} - H_{i,j-1}^{k}\right) - (C_\eta)_{i,j+1/2}^{k}\left(H_{i,j}^{k} - H_{i,j+1}^{k}\right)
\end{aligned}
\qquad (4.26b)
$$

The pentadiagonal matrix system for Equation 4.25 is shown in Figure 4.5. The matrix *A* is symmetric, banded, and sparse.

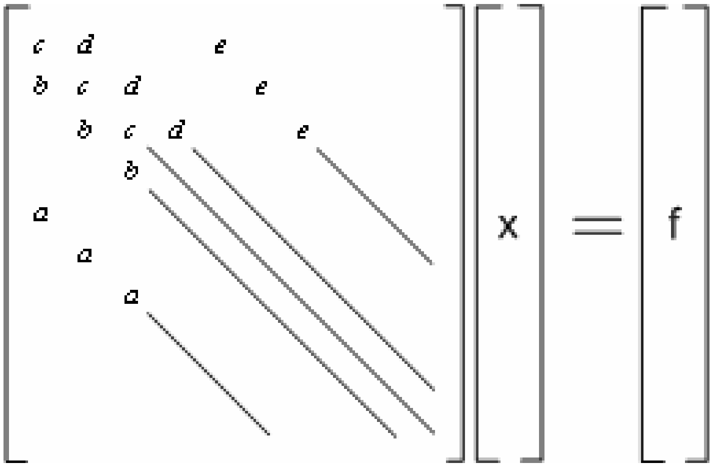


Figure 4.5: Pentadiagonal matrix systems

As shown in Equations 4.15 and 4.20, the indices of the coefficient matrix (which actually are conductance terms) are dependent on the solution vector of the system, making the system nonlinear. Strictly, the coefficient matrix cannot be defined unless the unknown vector is solved. Linearization is necessary to solve a nonlinear problem. Newton's method is known as the master method for solving nonlinear equations due to fast convergence speed. However, the cost of generating the Jacobian of transformation and solving the system by Gauss elimination may be high so that the overall cost is even greater than that of other iterative methods (Ferziger and Peric, 2002). A Picard iteration method used by Feng and Molz (1997) is used for the linearization. The nonlinear terms at the time level *k+1* are linearized by approximating the conductance terms with the solutions at the time level *k*. Then, the system of linear equations is solved and the coefficient matrix is updated with new solutions at *k+1* level. The solution of the next nonlinear iteration step (*m+1*) is compared to the previous nonlinear iteration step (*m*) solutions for the convergence check until the error converges enough. The solution process of the nonlinear iteration is as follows:

(1) At time level *k+1,* conductance terms are computed with *k* level solution. e.g. $\left(C_{\xi}\right)_{i+1/2,j}^{k}$

(2) The matrix system is solved for the vector *x*. e.g. $h_{i,j}^{k+1,m}$

(3) Update the matrix *A*. e.g. $\left(C_{\xi}\right)_{i+1/2,j}^{k+1,m+1}$.

(4) Update solution vector. e.g. $h_{i,j}^{k+1,m+1}$

(5) Check the convergence of the solution. e.g. L_2 norm$<10^{-5}$, L_∞ norm $<10^{-5}$

(6) Repeat (3), (4) and (5) until the solution converges

(7) Go to next time level

The number of iterations for this linearization depends on the time interval and the rate of change in the conductance terms between time steps. In this model, the time series solution is computed for a rising hydrograph. The algorithm adopted from the General Conjugate Gradient method preconditioned with the Modified Incomplete Cholesky Decomposition (MICCG) proposed by Cooley (1992) solves the system of equations iteratively. The solution process starts by creating a computational grid space based on the input road surface geometry. Boundary conditions are updated at each time step, and the MICCG solver iterates until current time step solution converges. At each time step, nonlinear diffusion coefficients are found by iterating the inner loop of MICCG solver. The solution process repeats as the time step advances until the flow reaches a steady state condition. Figure 4.6 outlines this procedure through a flow chart.

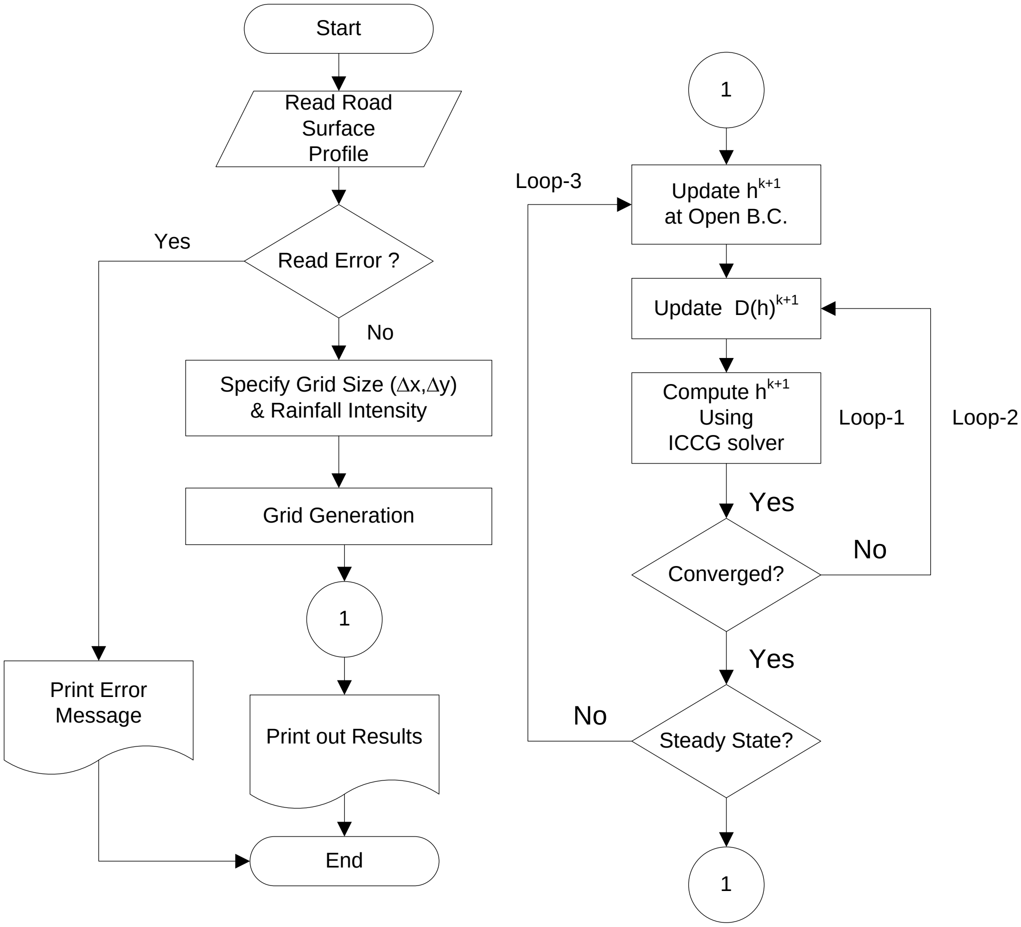


Figure 4.6: Solution Process

4.6 STABILITY

A numerical solution method is said to be stable if the computational errors in the numerical solution do not grow. The most widely used technique to check the stability of numerical schemes that solve linear problems is the von Neumann's method (Ferziger and Peric, 2002). For nonlinear problems such as Equation 4.11, a local stability analysis may provide a necessary condition for stability (Hirsch, 1988). In this analysis we use frozen values for the non-linear and non-constant coefficients to make the formulation linear. Thus, the result of the analysis will be a necessary condition but not sufficient for stability of the non-linear problem. The 2D diffusion equation for the proposed model is shown in Equation 4.11. Let us set up the discretized equation as a function of total head rather than water depth for simplicity.

$$\begin{aligned} & H_{i,j}{}^{k+1} - H_{i,j}{}^{k} \\ & = \sigma_x \left(H_{i+1,j}{}^{k+1} - 2H_{i,j}{}^{k+1} + H_{i-1,j}{}^{k+1} \right) + \sigma_y \left(H_{i,j+1}{}^{k+1} - 2H_{i,j}{}^{k+1} + H_{i,j-1}{}^{k+1} \right) \\ & + \sigma_x \left(H_{i+1,j}{}^{k} - 2H_{i,j}{}^{k} + H_{i-1,j}{}^{k} \right) + \sigma_y \left(H_{i,j+1}{}^{k} - 2H_{i,j}{}^{k} + H_{i,j-1}{}^{k} \right) + r \end{aligned} \tag{4.27}$$

where σ_x, σ_y are defined by

$$\sigma_x = \frac{D_x \Delta t}{2\Delta x^2}, \qquad \sigma_y = \frac{D_y \Delta t}{2\Delta y^2} \tag{4.28}$$

The diffusion coefficients are assumed to be constant. The discrete Fourier decomposition is defined by

$$H_{i,j}^{k} = U^{k} e^{Ipi\Delta x} e^{Iqj\Delta y} \tag{4.29}$$

where I is the imaginary unit, p and q are the wave numbers in the x and y directions, respectively, i.e., $\lambda_x = 2\pi/p$, where λ_x is the wavelength. U^k is the amplitude at time level k. Substituting this expression into Equation 4.25 gives

$$U^{k+1} = U^{k} + \sigma_x U^{k+1} \left(e^{Ip\Delta x} - 2 + e^{-Ip\Delta x} \right) + \sigma_y U^{k+1} \left(e^{Iq\Delta y} - 2 + e^{-Iq\Delta y} \right)$$

Divide the whole equation by $U^k e^{Ipi\Delta x} e^{Iqj\Delta y}$ and use the following relationship to get an expression of G.

$$\frac{\left(e^{Ip\Delta x} + e^{-Ip\Delta x}\right)}{2} = \cos(p\Delta x)$$

Introducing the an amplification factor such that $U^{k+1} = GU^k$, and rearranging the equation for the amplification factor, we get

$$G = \frac{1}{1 + 4\sigma_x \sin^2\left(\frac{p\Delta x}{2}\right) + 4\sigma_y \sin^2\left(\frac{q\Delta y}{2}\right)} \tag{4.30}$$

The numerical solution is stable only when |G|≤ 1. This condition applies to Equation 4.30 only when the following condition is accepted.

$$\sigma_x \sin^2\left(\frac{p\Delta x}{2}\right) + \sigma_y \sin^2\left(\frac{q\Delta y}{2}\right) \geq 0 \tag{4.31}$$

Since σ_x and σ_y are both positive and sine terms are at least equal to zero, the stability condition satisfies any condition. Thus, the numerical scheme of the proposed model is unconditionally stable. However, one should note that not only this unconditional stability requirement is satisfied locally but also the effect of the boundary condition on the stability of the solution is not included with this procedure. Thus the actual stability requirement may be more restrictive than the one obtained here. Nevertheless, the results still provide useful information on stability requirements.

In running the proposed model, there is no stability related issue appeared. The number of nonlinear iteration may increase if time interval is too large (or Courant number is large.)

4.7 EVALUATION OF MODEL CONVERGENCE

Theoretically the numerical solution x_i^k should approach the exact solution of the differential equation when the spatial and temporal intervals tend to zero. For a well-posed and consistent discretization scheme, stability is necessary and sufficient condition for convergence (Hirsch, 1998). From the practical point of view, it is important to specify appropriate stopping criteria to guarantee the solution to be reasonable. Vector norms are frequently used to check the convergence of iterative methods. There are three loops that iterate the solution procedure. The first inner most DO-loop iterates to seek for the solutions for any given condition using the CG solver. The second loop compares the linearized model solutions to get nonlinear solution at each time step. The last outer most loop stops only when the solution becomes steady state. L_2 norm and L_∞ norm will be used as stopping criteria in the model. To eliminate the 'order of magnitude' related truncation errors, the norms are normalized by the solution itself such that

$$\begin{aligned} \|x\|_2^{\,n} &= \frac{\sqrt{\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\left|\left(h_{i,j}^{\,k}\right)^n - \left(h_{i,j}^{\,k}\right)^{n-1}\right|^2}}{\left(h_{avg}^k\right)^n} \\ \|x\|_\infty^{\,n} &= \max_{i,j}\left|\frac{\left(h_{i,j}^{\,k}\right)^n - \left(h_{i,j}^{\,k}\right)^{n-1}}{\left(h_{i,j}^{\,k}\right)^n}\right| \end{aligned} \tag{4.32}$$

where n represents the n^{th} iteration and k the time step. In the loop for nonlinearity, n represents the number of nonlinear iteration. Therefore, the actual number of iterations becomes $n_{inner} * n_{outer}$. The difference in solutions between the current and previous iterations is compared and the process iterates until the norms becomes smaller than the stopping criteria. The solution is considered to be converged only if (1) when the difference in the solution between iterations is small (L_2 norm) and (2) the residual at current iteration is small (L_∞ norm).

$$\|x\|_2^n < \varepsilon_2 \text{, and } \|x\|_\infty^n < \varepsilon_\infty \tag{4.33}$$

In Equation 4.33 the parameters ε_2 and ε_∞ are the stopping criteria with magnitudes such as 10^{-5}. Figure 4.7 shows the convergence of the numerical solution of the CG solver. Both L_2 and L_∞ norms decrease exponentially as the number of iteration increases during an iterative solution process.

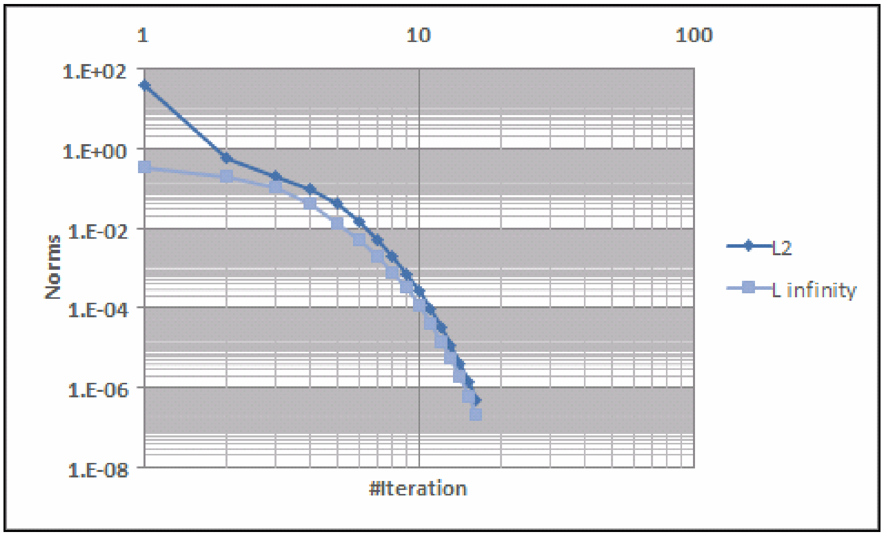
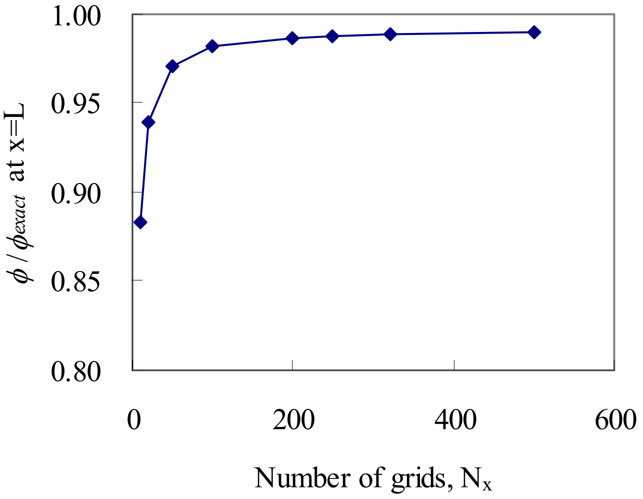


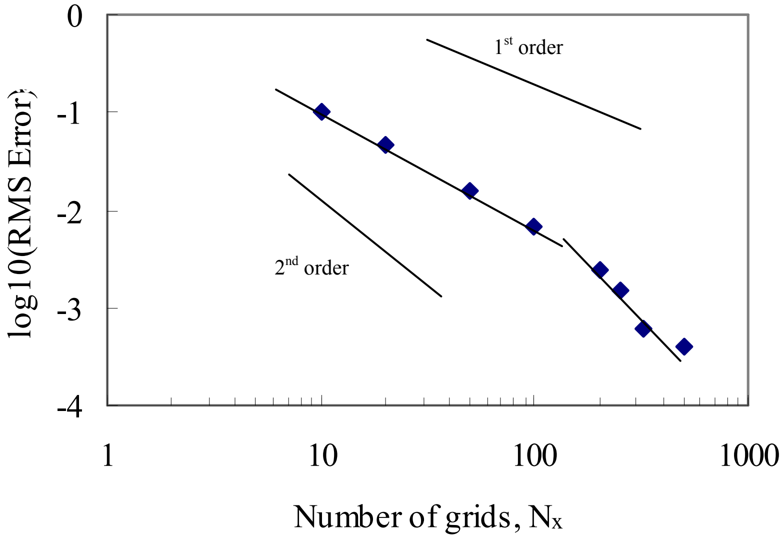
Figure 4.7: Convergence speed of MICCG solver measured by L_2 and L_∞ norms.

A problem with one-dimensional flow is simulated with the following variables: $L = 0.6$m, $r = 250$mm/hr, and $S_x = 1\%$. Since the proposed numerical model is built for 2D simulations, a true 1D simulation is not available. Therefore, a flow with only longitudinal component of velocity ($u_x \neq 0$, $u_y=0$) is defined as 1D flow. The solution with N_x=1000 (Δx=0.0006 m) is used as the exact solution. Figure 4.8(a) shows the variation of the water depth as the grid is refined for the central differencing scheme

(CDS). The grid was refined from $N_x = 10$ to $N_x = 500$. On the coarsest grid, the model does not produce a meaningful solution. As the grid is refined, the model result converges monotonically towards a grid-independent solution.



(a) Convergence of downstream water depth



(b) Error in computed water depth

Figure 4.8: Convergence and errors in 1D simulation

The normalized RMSE errors are plotted with respect to the normalized grid size on the bottom of Figure 4.8(b). The expected first- and second-order are also shown. The slope of the error curve is expected a second-order as the model is written in CDS; however, the error shows irregular behavior on coarse grids with the slope of the first-order convergence. As the grid is refined, the error reduces in second-order. This result is fairly standard as the basic idea of error convergence is that, for a sufficiently fine space (or time) interval, the error should converge. However, it is well known that at sufficiently coarse space (or time) steps the underlying PDE is not well-represented and error will not converge at high order (Ferziger and Peric, 2002). It is likely that the influence of the boundary conditions reaches deeper into the domain at coarse grid resolutions, so that the error convergence is poorer. Furthermore, numerical stability of implicit methods may not guarantee the numerical accuracy at large Courant–Friedrichs–Lewy (CFL) numbers. Hodges (2004) suggested that the second-order accuracy of the Crank-Nicolson method for unsteady flows was not assured for CFL larger than unity. For example, if the error associated with space interval dominates the error of the model solution at large time intervals, the slope of error may not be of the second order as time interval decreases.

Similar behavior is observed in a 2D error simulation. An inclined rectangular domain is used for 2D simulation with the variables as follow: L = 30 m, W = 15 m, r = 250mm/hr, and $S_x = S_y = 0.01$. The steepest surface slope goes diagonally from the north-west corner to the south-east corner. In Figure 4.9, the error with respect to the number of grids is plotted in the log-log scale.

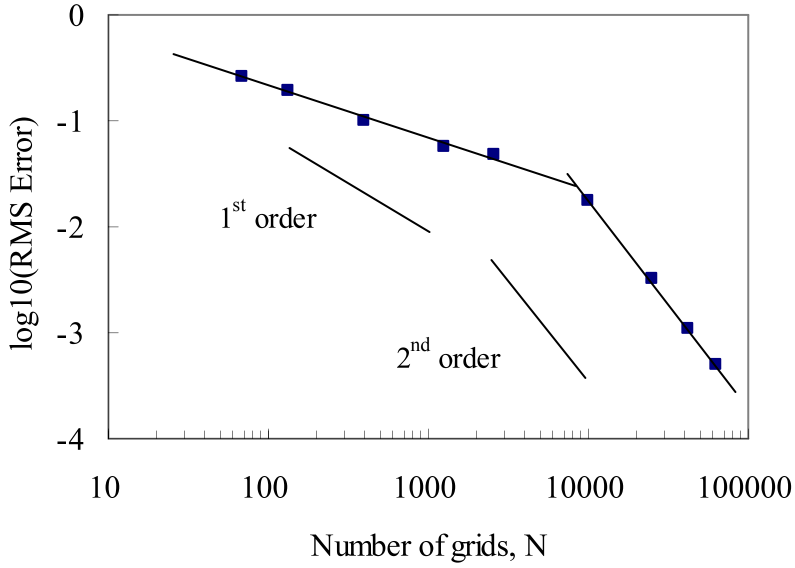
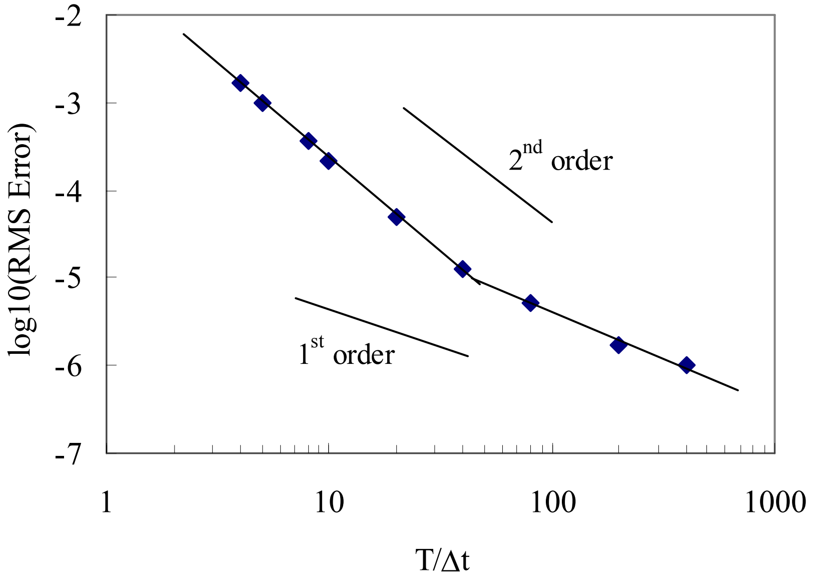


Figure 4.9: Errors in 2D simulation for different cell sizes

Ferziger and Peric (2002) suggests that the solution of an implicit method will be bounded if the time interval (*Δt*) is smaller than twice of the maximum *Δt* of an explicit scheme limited by CFL condition. Figure 4.10 shows the error in 1D flow simulation with respect to time interval. To assess the influence of nonlinear iteration, the model is modified such that diffusion coefficient is constant (top), then the original model is simulated for comparison (bottom).



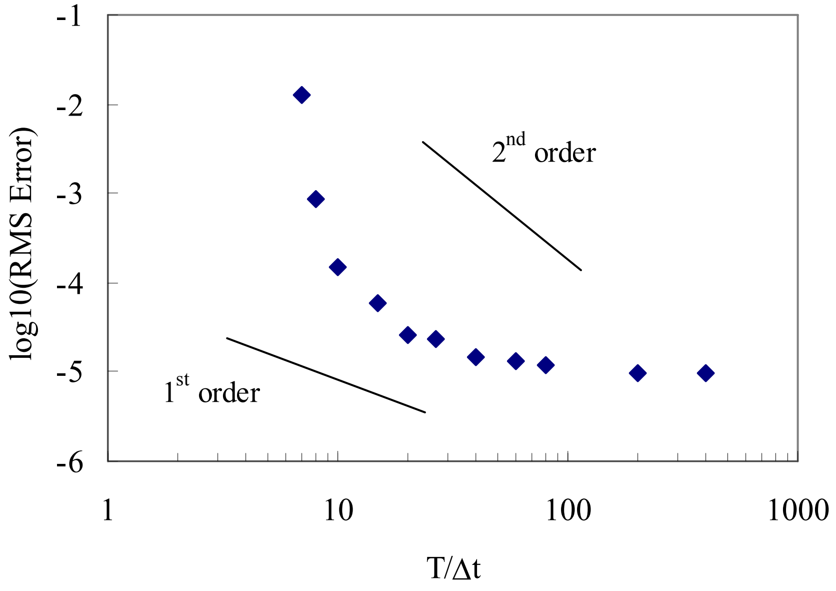


Figure 4.10: Error in the solution for 1D flow with different time intervals (top: linear model, bottom: nonlinear model)

The model solutions are all bounded within the simulated range of time intervals. The pattern of error in the linear model is piecewise linear: second-order on large time intervals and first-order on small intervals. Meanwhile, the pattern of error in the nonlinear model is similar to the linear model while it is more rounded in shape. Also, the rate of error reduction as a function of time interval is very slow at small time intervals. The error generated from nonlinear iteration or the error generated from relatively large space interval may cause the result not to be second-order of CN method as the resolution of time interval increases.

As time step increases the solution eventually reaches steady state. After then, any further simulation is trivial under constant rainfall intensity because the solution becomes constant. Therefore, the model stops running when the error between time steps becomes small. The L_2 and L_∞ norms are defined such that they compare the solutions between two time steps.

$$\begin{aligned} \|x\|_2^{\,k} &= \frac{\sqrt{\sum_{i=1}^{Nx}\sum_{j=1}^{N_y}\left|h_{i,j}^{\ k} - h_{i,j}^{\ k-1}\right|^2}}{h_{avg}^{k}} \\ \|x\|_\infty^{\,n} &= \max_{i,j}\left|\frac{h_{i,j}^{\ k} - h_{i,j}^{\ k-1}}{h_{i,j}^{\ k}}\right| \end{aligned} \tag{4.34}$$

When the L_2 and L_∞ norms become smaller than the stopping criteria, the solution is considered to have reached steady state. It should be noted that the stopping criteria need to be determined carefully so numerical errors do not propagate during the solution procedure. Since the norms are normalized by average numbers, tolerance values for CG solver and nonlinear iteration used in the model simulation are 0.001 for CG iteration and 0.005 for nonlinear iteration. These values balance the accuracy in the solution and the computing time most efficiently. Same criteria applies to L_2 and L_∞ norms such that the iteration continues until both norms become smaller than the tolerance values.

Computation time increases proportionally as the size of a domain increases. Average computation time for sheet flow to reach steady state on different widths of roads is presented in Table 4.1. The model is run on personal computers with Intel(R) Pentium(R) 4 CPU 2.80GHz and 1GB RAM. As the number of lanes doubles, the average computing time also increases nearly twice larger.

Table 4.1: Computation time with respect to number of lanes

Number of lanes*	Number of cells**	t_{avg} (min)***
4	26,220	37.1
6	35,343	51.7
8	44,460	65.0

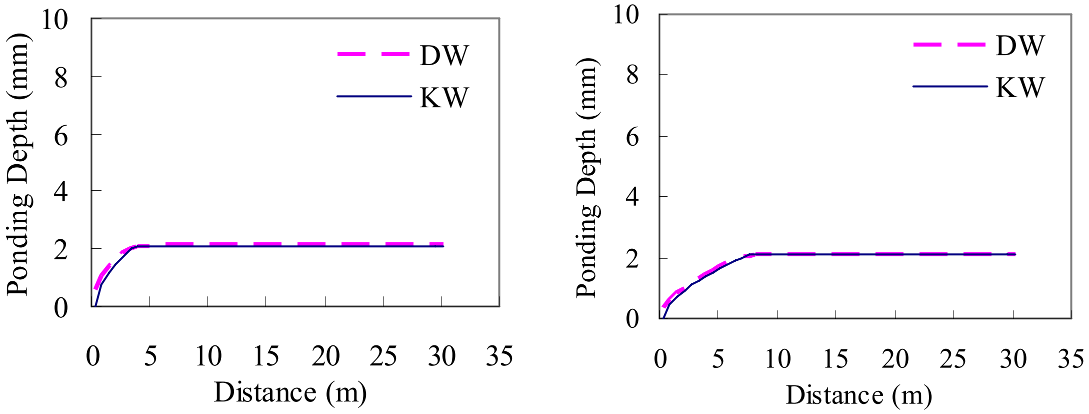
* numbers for both sides of a road

** L=230m, $W_{each\ lane}$=3.7m, and $W_{shoulder}$=3m, Δx=0.6m, Δy=0.3m, Δt=0.4sec

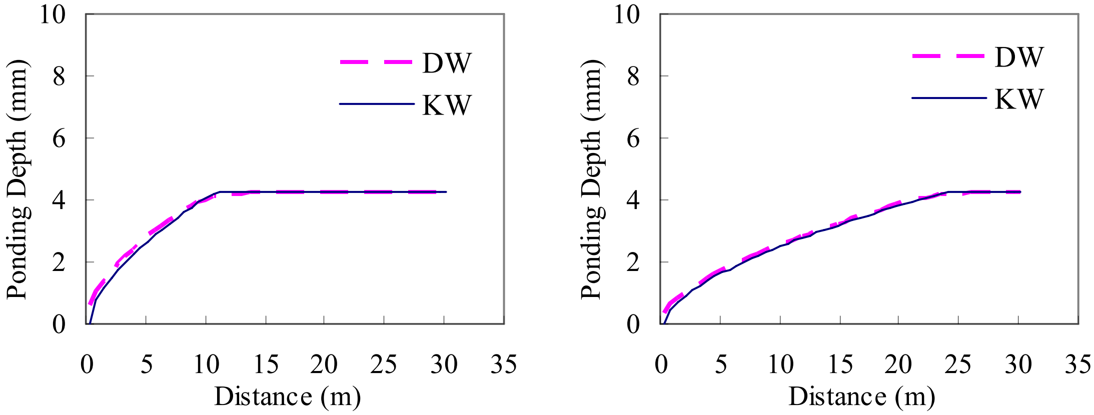
*** time to reach steady state condition

4.8 MODEL VERIFICATION

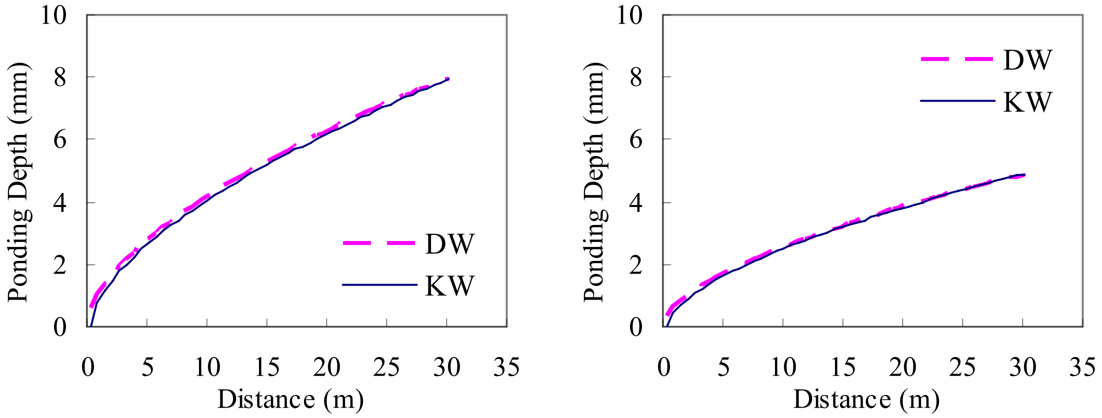
One advantage of the proposed computer model is versatility in application. As long as the geometry file is written such that the model can read, the proposed model generates a grid space and computes the sheet flow depth and discharge on the domain. In this chapter, the model solution of one dimensional flow will be compared to the kinematic wave model solution to validate the accuracy of the model. The model equation shown in Equation 4.11 does not have an analytical solution, so the numerical solution of the proposed model can not be directly compared to the exact solution for validation; instead, the model solution is compared with an analytical solution of the kinematic wave model for a case of 1D flow. A mild slope surface (S=1%) and a steep slope surface (S=5%) are used for the simulation. The results are shown in Figure 4.11.



(a) at t=30 seconds



(b) at t=60 seconds



(c) at steady state ($T_{c,KW}$ =120sec, $T_{c,DW}$ =70sec)

Figure 4.11: Comparison of the model solutions with kinematic wave model solutions at different time levels. r = 250mm/hr, n = 0.015, L = 30 m, Δx = 0.6 m, N_x = 50. (S_o = 1% shown on the left, 5% on the right)

The DW solutions are fairly accurate compared with the analytical KW solutions as shown in Figure 4.11. The RMS error of DW solution at steady state for 1% (left in Figure 4.11) and 5% (right) are 0.043% and 0.023%, respectively. The no-through flow boundary condition at the upstream end in the proposed diffusion wave model makes a difference in the steady state solution. Consequently, about 50% of the error occurs at the 5 grid points on the upstream end and the accuracy of the model solution is even better than it appears in the RMS errors.

Time series solutions are compared in Figure 4.12. The KW analytical solution and the DW numerical solution are in good agreement on a rising hydrograph. A smooth transition in DW solution near the peak is observed. As Kazezyılmaz-Alhan and Medina (2007) noted, the flow depth gradient term in the rising part of the DW model affects the solution being smooth near the peak. It is also observed that the DW model slightly overestimates the flow rate at the steady state compared to the kinematic wave solution. Based on the spatial and temporal comparisons, we can conclude that the proposed DW model gives reliable solutions for 1D flow and may be extended to 2D flow problems.

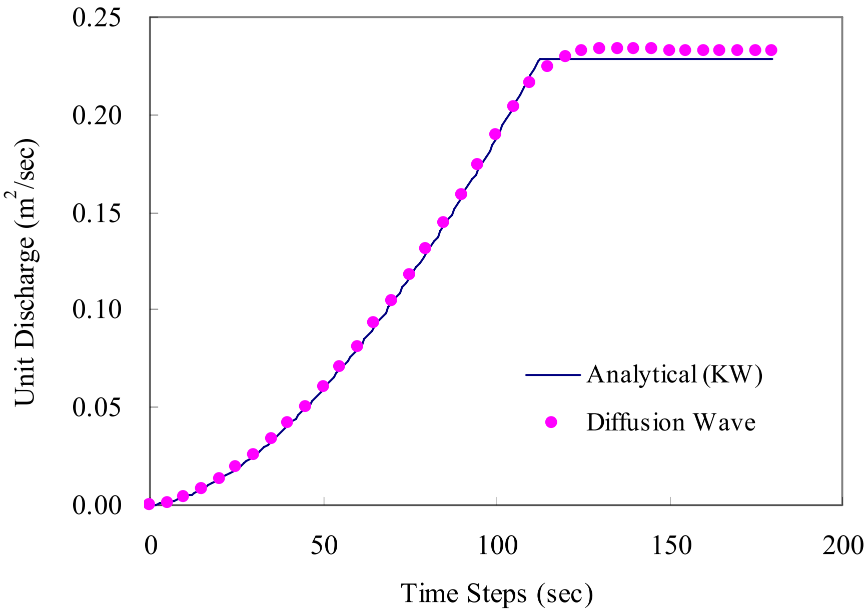


Figure 4.12: Rising hydrograph of the diffusion wave model for a 1D flow: $S_o = 1\%$, $n = 0.015$, $r = 200$ mm/hr, $L = 30$ m.

4.9 DISCUSSION ON FLOW MODELS

The relative magnitude of the terms in the momentum equation is important for choosing a hydrodynamic wave model. Henderson (1966) computed the relative magnitude of momentum equation terms for a steep alluvial channel with a fast rising hydrograph. He concluded that the discharge can be computed as in uniform flow (kinematic wave approximation) suggesting that the relative magnitude of bed slope is 100 times larger than the inertial terms in order of magnitude in the momentum equation. He also noted that the depth gradient term may well be of the same order as the bed slope when the bed slope is very flat. Yu and McNown (1964) considered use of Keulegan's equation for calculation of runoff from the end of a pavement section. They concluded that the gravity and bed resistance terms were larger than others by a factor of 100, and thus a kinematic model could be used based on the continuity equation and a resistance equation. Due to the simplicity in the formula, the computation cost and difficulty of kinematic wave models are significantly lower than full dynamic or diffusion wave models while the computed results are fairly accurate. A 1D sheet flow simulation computed with the kinematic wave model under a constant rainfall and surface slope at steady state is compared with the diffusion wave model result in Figure 4.13. The curve with circled marker represents the kinematic wave solution, and the triangle shows the diffusion wave solution. There is no apparent difference observed between the solutions.

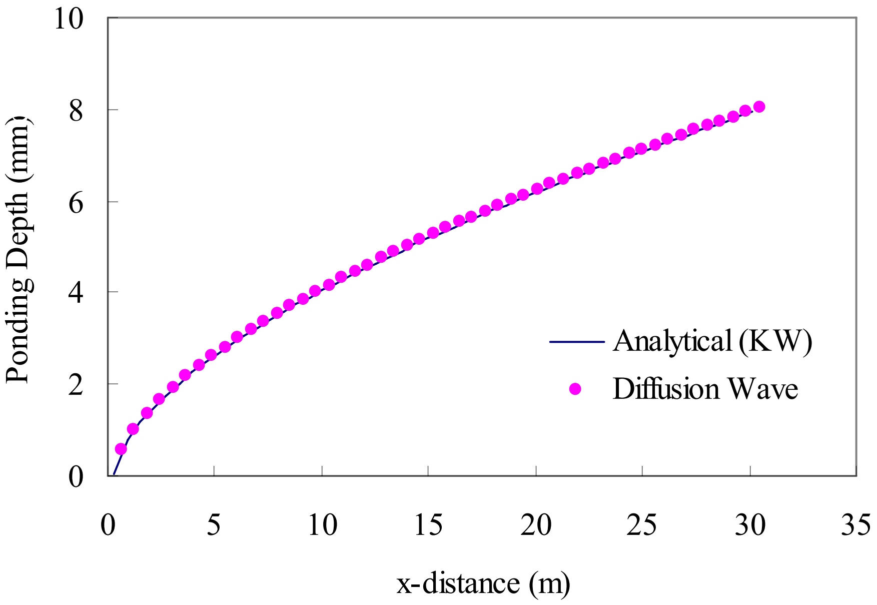


Figure 4.13: 1D flow under constant rainfall at steady state

However, the application of the kinematic wave model is limited for domains that have areas 1) where bed slope is zero or 2) where there is significant pressure gradient. Due to the fact that the surface slope is the only term that contributes to friction slope, zero slopes within a domain will result in discontinuity in the solution. Figure 4.14 shows a 2D sheet flow under constant rainfall with change in cross slope. A stream tube method with kinematic wave approximation was used to compute water depth at steady state. The streamlines head to the left side near the upstream, but the direction gradually changes to inside domain as they go downstream. There exists a clear discontinuity in the water depth between streamlines that barely touch the left side boundary forming a long flow path, and streamlines that begin on the side of the domain. As streamlines head back to the domain, new streamlines begin at the nearby side boundary. If surface slope is the only term in the equation of motion, the difference of water depth on these streamlines can not be counted appropriately resulting in a sharp decrease in water depth.

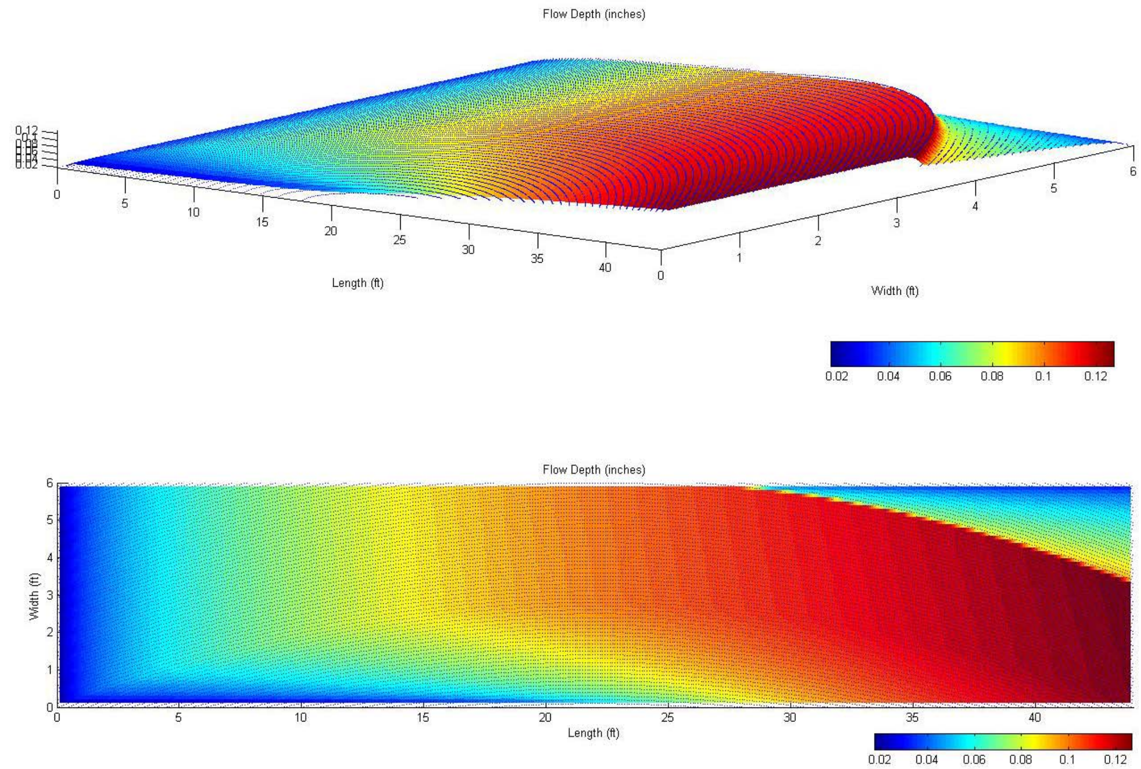


Figure 4.14: Kinematic wave solution for a 2D flow

The flow pattern shown in Figure 4.14 is typical on a superelevation transition area which means the head gradient term in the momentum equation can not be neglected. Therefore, the kinematic wave model is not adequate for the domains with this kind of irregular topography. As a result, a diffusion wave model is considered to be the one that gives a fairly accurate solution without a discontinuity contributed by topography, and the one that significantly saves numerical computation cost compared with a full dynamic wave model.

4.10 INITIAL CONDITION AND BOUNDARY CONDITIONS

4.10.1 Initial Condition

It is assumed that there is no ponding of water on the domain before rain starts. Therefore a ‘dry’ initial condition is set for the model,

$$h_{i,j}^{0} = 0 \tag{4.35}$$

Equation 4.35 is equivalent to stating that the initial water surface elevation is the same as the elevation of the pavement surface, $H_{i,j}^{0} = z_{i,j}$.

4.10.2 Kinematic Boundary Conditions

Kinematic boundary conditions are applied using the method of characteristics (MOC) on both the upstream and downstream ends of the grid where the roadway surface has a normal crown or is plane and in superelevation. Outflow kinematic boundary conditions are also applied along the sides of the roadway where no curbs are present. The methods of implementation for inflow and outflow boundary conditions are different. In one case, the characteristic curves approach the boundary from outside of the domain. In the second case, the characteristic curves approach the boundary from within the domain. These two cases are shown in Figure 4.15.

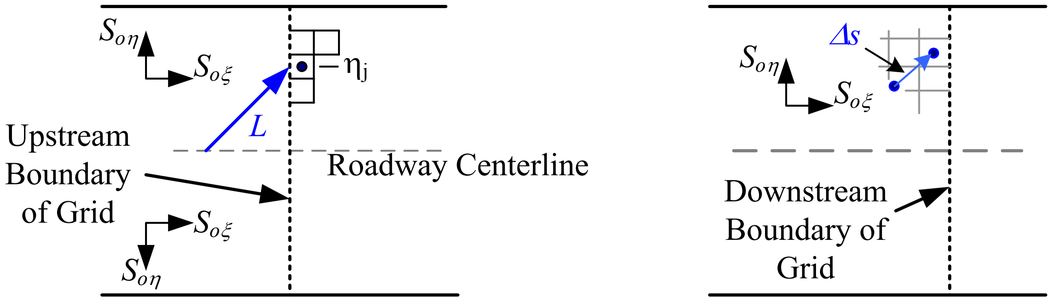


Figure 4.15: Implementation of kinematic boundary conditions when characteristics arrive at the boundary from outside (left) of the domain and within (right) the domain

First consider the case shown to the left of Figure 4.15. The base characteristic that reaches the domain boundary at location η_j leaves the roadway centerline (crown) at a location determined by the pavement longitudinal and cross slope $S_{o\xi}$ and $S_{o\eta}$. Recalling that the roadway centerline corresponds to η = 0.5 and that the length scale ratio from Equation 3.15 equals the width of the roadway (W), the drainage path length (L_d) from the roadway crown to the boundary is equal to

$$L_d = \left|\eta_j - 0.5\right| W \frac{S_o}{S_{o\eta}} \tag{4.36}$$

With this drainage path length and a constant rainfall rate starting at time zero, the unit flux at the grid boundary reaches its maximum value at the time of concentration for this drainage path. These values of maximum unit discharge and time of concentration are given by

$$q_{\max}(\eta_j) = r\left|\eta_j - 0.5\right| W \frac{S_o}{S_{o\eta}} \tag{4.37}$$

$$T_c = \left(\frac{nL_d}{\phi\sqrt{S_o}}\right)^{0.6}\left(\frac{1}{r}\right)^{0.4} \tag{4.38}$$

The MOC equations provide the depth and unit flux at the boundary for time $t < T_c$.

$$h_B(\eta_j, t) = r\,t \quad ; \quad t \le T_c \tag{4.39}$$

$$q_B(\eta_j, t) = \frac{\phi\sqrt{S_o}}{n}(r\,t)^{5/3} \quad ; \quad t \le T_c \tag{4.40}$$

The flow rate is applied as a Neumann boundary condition through an incremental addition to the rainfall intensity of the boundary cell. The normal component of the flow across the boundary is calculated as q_B $(S_{o\xi}/S_o)$. Assuming that the boundary corresponds to the first column (i = 1) and that the cell column width is $\Delta\xi_1$, then the incremental rainfall added to the boundary cell is

$$\Delta r_{1,j} = \frac{q_B(\eta_j, t)}{L_\xi(\xi_1, \eta_j)\Delta\xi_1}\frac{S_{o\xi}}{S_o} \tag{4.41}$$

Figure 4.16 shows a plot of incremental rainfall on the upstream boundary for half of a normal crowned roadway (center to side end). The variables are normalized by largest values. As η increases the time of concentration also increases. Once the time reaches T_c, the rainfall loading becomes steady (Equation 4.40). Thus, the steady state for the incremental rainfall loading on the upstream boundary begins from the roadway center (η=0.5) at t=0 (theoretically $L_{d,center}$=0) and propagates to the side of the roadway (η=1).

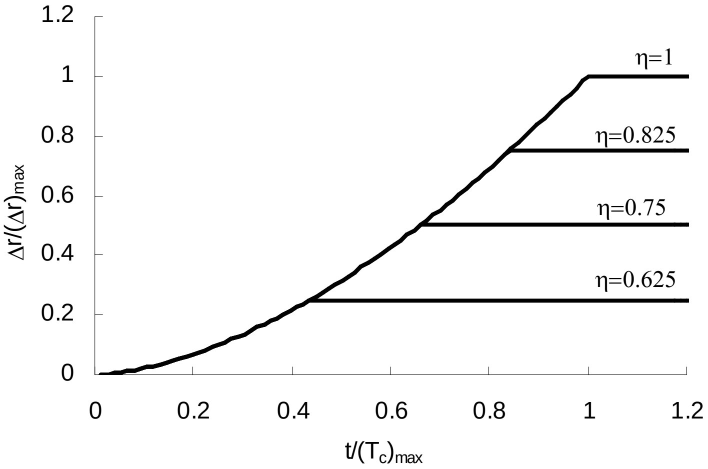


Figure 4.16: Incremental rainfall loading on the upstream boundary. ($S_{o\xi}$= 2%, $S_{o\eta}$= 2%, W = 7.3m, $\Delta x = \Delta y$ = 0.3m, r = 125mm/hr, n = 0.015)

For the case shown on the right side of Figure 4.15, the characteristic curve arrives at the boundary from within the domain. For this case, the objective is to apply the kinematic boundary condition as a Dirichlet condition at time level k+1 with depth interpolated from the depth values at time level k and updated using the characteristic equations. The algorithm is best presented in a general setting with point 2 downstream of initial point 1 along a characteristic curve. For point 2, located at the center of a boundary cell, h_2^{k+1} can be predicted given the numerical solution at time level k. Since the roadway surface is plane and uniform, the kinematic wave model is the one-dimensional form of Equation 4.1 (Henderson and Wooding, 1964). The MOC solution can be formulated either in terms of depth or unit discharge. For each formulation the MOC solution gives

$$h = h_2^k + r\tau \tag{4.42}$$

$$q = q_1^{\,k} + r\left(s - s_1\right) \tag{4.43}$$

In equations 4.42 and 4.43, τ refers to time ($\tau = t - t^{\,k}$) and s to distance along a characteristic curve. The selection of 'initial conditions' is determined through the following discussion. If $c(h) = dq/dh$ is the characteristic celerity, then the base characteristic equation gives

$$\frac{ds}{d\tau} = c\left(h\right) = \frac{dq}{dh} = \frac{5}{3}\frac{\sqrt{S_o}}{n}\,h^{2/3} \tag{4.44}$$

Combining Equations 4.42 and 4.44 and integrating the resulting form over the time increment ($\Delta t = t^{\,k+1} - t^{\,k}$) give

$$\Delta s = \frac{\sqrt{S_o}}{n\,r}\left[\left(h_2^{\,k} + r\Delta t\right)^{5/3} - \left(h_2^{\,k}\right)^{5/3}\right] \tag{4.45}$$

Equation 4.45 gives the distance moved along the base characteristic *B* shown in Figure 4.17 during the time increment $t^{\,k} \rightarrow t^{\,k+1}$. The increase in unit discharge along this flow path is $\Delta q = r\Delta s$. It is assumed that the same distance is traversed along the characteristic *A* shown in Figure 4.17, and use of Equation 4.43 results in the following algorithm (when combined with the Manning equation)

$$h_2^{\,k+1} = \left(\left(h_1^{\,k}\right)^{5/3} + \left(h_2^{\,k} + r\Delta t\right)^{5/3} - \left(h_2^{\,k}\right)^{5/3}\right)^{0.6} \tag{4.46}$$

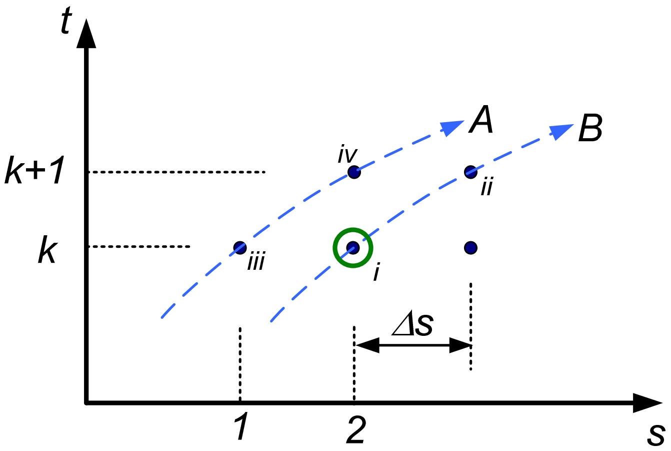


Figure 4.17: Schematic view of computational grid for outflow kinematic boundary condition

Specifically, with reference to Figure 4.17, the point correspondence is $i \rightarrow h_2^k$, $ii \rightarrow h_2^k + r\Delta t$, $iii \rightarrow h_1^k$, and $iv \rightarrow h_2^{k+1}$. The algorithm is applied by first using Equation 4.45 to determine the distance upstream from point 2 to point 1, then interpolates between cell values using the numerical solution at time t^k, and finally uses the interpolated value as h_1^k in Equation 4.46 to determine the Dirichlet boundary value for the boundary cell.

The open boundary conditions at upstream end and downstream ends are evaluated by simulating a plane tilted surface. The upstream and downstream boundary values are compared with those within the domain. For this comparison, cell values at the same location in the transverse direction are compared. A diagonally slanted asphalt surface with L=30m, W=15m, with S_y=2.0%, r=250mm/hr is used in this test. Since the surface slope is constant everywhere, the ponding depth should be constant in the longitudinal direction at any cross section within the domain including boundaries. For the comparison, steady state solution at a station in the middle of domain is used as reference (i.e. exact solution).

Table 4.2: Errors in the upstream boundary condition

S_x (%)	0.1	1.0	2.0	4.0
RMS (mm)	0.023	0.064	0.072	0.079
$\lvert d\rvert_{max}$[a] (mm)	0.073	0.122	0.169	0.216
$\lvert d\rvert_{min}$ (mm)	0.000	0.000	0.000	0.000
$\lvert d\rvert_{avg}$ (mm)	0.021	0.058	0.063	0.059

[a] difference in upstream boundary values between those estimated by the open B.C and exact solution

As shown in Table 4.2 the error of the upstream boundary condition increases as the longitudinal slope increases. Extended flow paths which contribute to the inflow to upstream end result in the increase in error. However, the estimated error is overall very small such that the error is negligible. The estimated error in the downstream boundary condition is summarized in Table 4.3. Again, the estimated error is small enough to be ignored. No trend is found in the error with respect to longitudinal slope.

Table 4.3: Errors in the downstream boundary condition

S_x (%)	0.1	1.0	2.0	4.0
RMS (mm)	0.018	0.005	0.004	0.029
$\lvert d\rvert_{max}$[a] (mm)	0.034	0.026	0.018	0.057
$\lvert d\rvert_{min}$ (mm)	0.000	0.000	0.000	0.000
$\lvert d\rvert_{avg}$ (mm)	0.017	0.003	0.002	0.019

[a] difference in downstream boundary values between those estimated by the open B.C and exact solution

4.10.3 No-flow (Neumann) Boundary Conditions

Highways with curbs have closed boundaries on the roadsides. A no-through flow boundary condition applies to these boundaries. In this boundary condition, water surface is forced to be flat by defining the zero-gradient in total head near the boundary.

$$\frac{\partial H_{i,j}^{n}}{\partial \eta} = 0; \quad j=1,\, N_y \tag{4.47}$$

This boundary condition also holds for curb-less roadside where the cross slope is uphill toward the boundary so water flows from the boundary toward the road center. Another Neumann-type B.C. is zero water-depth gradient condition. This boundary condition defines the normal flow condition where water depth is constant through the boundary. For cases with roadside curbs, sheet flow becomes channelized along the low side of the road (gutter flow). The algorithm automatically detects the grid points on the downstream boundary on which the flow is kinematic or the water surface is flat due to the channelization. Then either zero-depth gradient B.C or kinematic B.C is selectively applied on the downstream boundary cells.

4.11 ALGORITHM FOR CURB-OPENING INLETS

This subsection describes algorithms that have been developed to simulate flow near curb-opening inlets. Curb-opening inlets are commonly installed along the curb side of highways. The interval between inlets and the inlet size are determined based on the amount of stormwater runoff for a certain design rainfall intensity. A typical curb-opening inlet configuration is shown in Figure 4.18.

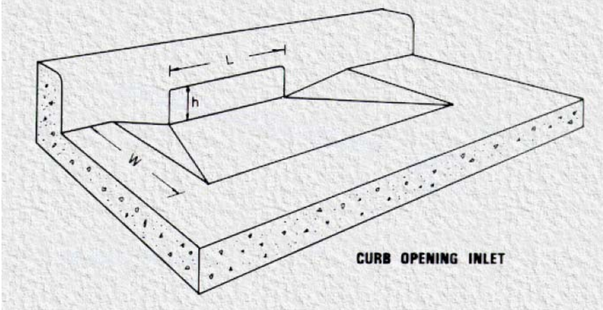


Figure 4.18: A depressed curb-opening inlet (HEC-12).

A curb-opening inlet places the inlet on the wall of the roadside curb; therefore, it is reasonable to design an algorithm such that the inlet is located at the outside of roadside boundary cells. The schematic at the cell scale is shown in Figure 4.19. The critical flow condition applies on the outside interface of a boundary cell. Thus, the proposed algorithm does not consider lateral flooding over inlets.

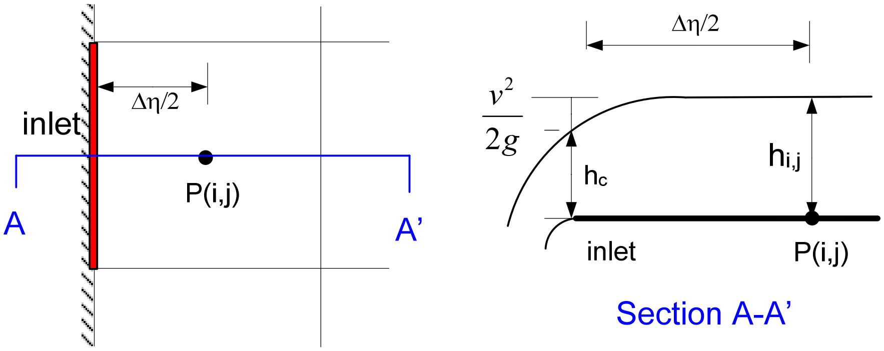


Figure 4.19: Cell scale configuration of a curb-opening inlet

For the critical flow condition, the Froude number is forced to be one. Then one gets a relationship between flow rate and water depth.

$$F = \frac{v}{\sqrt{gh}} = 1 \quad \Rightarrow \qquad v_c = \sqrt{gh_c} \tag{4.48}$$

Assuming that the flow velocity at the cell center corresponds to the available specific energy for lateral flow into the curb inlet, one may use the critical flow condition at the inlet to determine the inlet depth as a function of flow depth at the center of the cell (at critical flow, one-third of the specific energy is associated with flow velocity and two-thrirds is assicoated with flow depth.)

$$\left(h_c^{k+1}\right)^{m+1} = \frac{2}{3}\left(h_{i,j}^{k+1}\right)^{m} \tag{4.49}$$

In Equation 4.49 the index m denotes the m^{th} step of the nonlinear iteration. The critical depth at the current nonlinear iteration step is estimated based on the values computed at the previous iteration. The diffusion coefficient for the flow coming into the inlet opening is estimated as follows.

$$\left(D_\eta\right)_{inlet} = \frac{h_c^{5/3}}{n\sqrt{S_f}} \quad \text{where} \quad S_f = \frac{H_{i,j} - \left(z_{inlet} + h_c\right)}{w_{i,j}/2} \tag{4.50}$$

In Equation 4.50 $w_{i,j}$ is the width of the cell i,j. The conveyance on the inlet boundary is

$$\left(C_\eta\right)_{inlet} = \frac{2\left(D_\eta\right)_{inlet}}{w_{i,j}^{2}} \tag{4.51}$$

There are two possible scenarios in this case concerning the placement of the inlet.

Case I. Inlet is located at *(i,1)*. The discrete equation for this case takes the form:

$$\begin{aligned}\Big(\left(C_\xi\right)_{i-1/2,1} + \left(\boldsymbol{C_\eta}\right)_{\boldsymbol{inlet}} + \left(C_\xi\right)_{i+1/2,1} + \left(C_\eta\right)_{i,1+1/2} + 2/\Delta t\Big)h_{i,1}^{k+1} - \left(C_\xi\right)_{i-1/2,1} h_{i-1,1}^{k+1} \\ - \left(C_\xi\right)_{i+1/2,1} h_{i+1,1}^{k+1} - \left(C_\eta\right)_{i,1+1/2} h_{i,2}^{k+1} = f_{i,1}\end{aligned} \tag{4.52}$$

where

$$
\begin{aligned}
f_{i,1} &= (2/\Delta t)h_{i,1}^{k} + 2r \\
&- \left(C_\xi\right)_{i-1/2,1}^{k+1}\left(z_{i,1} - z_{i-1,1}\right) - \left(C_\xi\right)_{i+1/2,1}^{k+1}\left(z_{i,1} - z_{i+1,1}\right) \\
&- \left(C_\eta\right)_{inlet}^{k+1}\left(z_{i,1} - z_{inlet}\right) - \left(C_\eta\right)_{i,1+1/2}^{k+1}\left(z_{i,1} - z_{i,2}\right) + \left(C_\eta\right)_{inlet}^{k+1} h_{inlet}^{k+1} \\
&- \left(C_\xi\right)_{i-1/2,1}^{k}\left(H_{i,1}^{k} - H_{i-1,1}^{k}\right) - \left(C_\xi\right)_{i+1/2,1}^{k}\left(H_{i,1}^{k} - H_{i+1,1}^{k}\right) \\
&- \left(C_\eta\right)_{inlet}^{k}\left(H_{i,1}^{k} - H_{inlet}^{k}\right) - \left(C_\eta\right)_{i,1+1/2}^{k}\left(H_{i,1}^{k} - H_{i,2}^{k}\right)
\end{aligned}
$$

Case II. Inlet is on the side of (*i,N$_y$*)

$$
\begin{aligned}
&\left(\left(C_\xi\right)_{i-1/2,Ny} + \left(\boldsymbol{C_\eta}\right)_{\boldsymbol{inlet}} + \left(C_\xi\right)_{i+1/2,Ny} + \left(C_\eta\right)_{i,Ny-1/2} + 2/\Delta t\right)H_{i,Ny}^{k+1} - \left(C_\xi\right)_{i-1/2,Ny} H_{i-1,Ny}^{k+1} \\
&\quad - \left(C_\xi\right)_{i+1/2,Ny} H_{i+1,1}^{k+1} - \left(C_\eta\right)_{i,Ny-1/2} H_{i,Ny-1}^{k+1} = f_{i,Ny}
\end{aligned}
\tag{4.53}
$$

where

$$
\begin{aligned}
f_{i,N_y} &= (2/\Delta t)h_{i,1}^{k} + 2r \\
&- \left(C_\xi\right)_{i-1/2,N_y}^{k+1}\left(z_{i,N_y} - z_{i-1,N_y}\right) - \left(C_\xi\right)_{i+1/2,N_y}^{k+1}\left(z_{i,N_y} - z_{i+1,N_y}\right) \\
&- \left(C_\eta\right)_{i,N_y-1/2}^{k+1}\left(z_{i,N_y} - z_{i,N_y-1}\right) - \left(C_\eta\right)_{i,inlet}^{k+1}\left(z_{i,N_y} - z_{inlet}\right) + \left(C_\eta\right)_{inlet}^{k+1} h_{inlet}^{k+1} \\
&- \left(C_\xi\right)_{i-1/2,N_y}^{k}\left(H_{i,N_y}^{k} - H_{i-1,N_y}^{k}\right) - \left(C_\xi\right)_{i+1/2,N_y}^{k}\left(H_{i,N_y}^{k} - H_{i+1,N_y}^{k}\right) \\
&- \left(C_\eta\right)_{inlet}^{k}\left(H_{i,N_y}^{k} - H_{inlet}^{k}\right) - \left(C_\eta\right)_{inlet}^{k}\left(H_{i,N_y}^{k} - H_{inlet}^{k}\right)
\end{aligned}
$$

The following example is considered: A 3-meter long curb inlet is placed at $d_{in} = 3$ m from the zero cross slope section as shown in Figure 4.20. Three different surfaces with 2%, 4%, and 6% longitudinal slopes are simulated. After 120 seconds, the profile of water depth in the transverse direction at the station where the curb-opening inlet is placed shows depression of water surface near the curb especially on the right side in Figure 4.21.

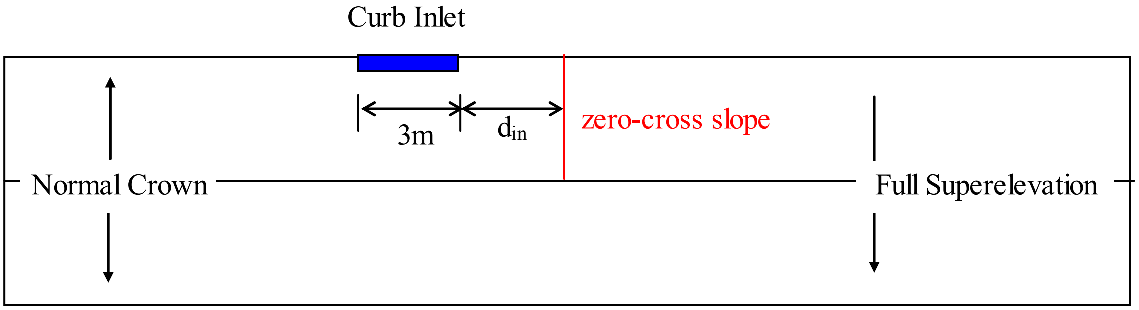


Figure 4.20: Scenario for curb-opening inlet simulation

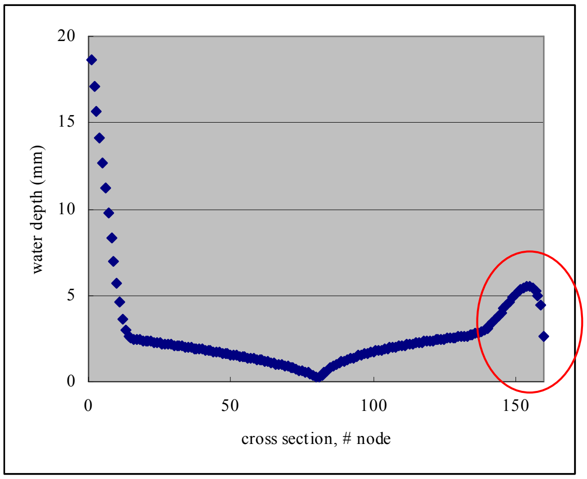


Figure 4.21: Depression in ponding depth at the inlet (node 160)

The longitudinal profile of water depth along the road direction at the center of the left half of the road near the curb-opening is shown in Figure 4.22. Water depth is large for mild slope and decreases as the slope increases. The existence of curb inlet resulted in decreasing the water depth on traffic lanes by 1 ~ 2 mm.

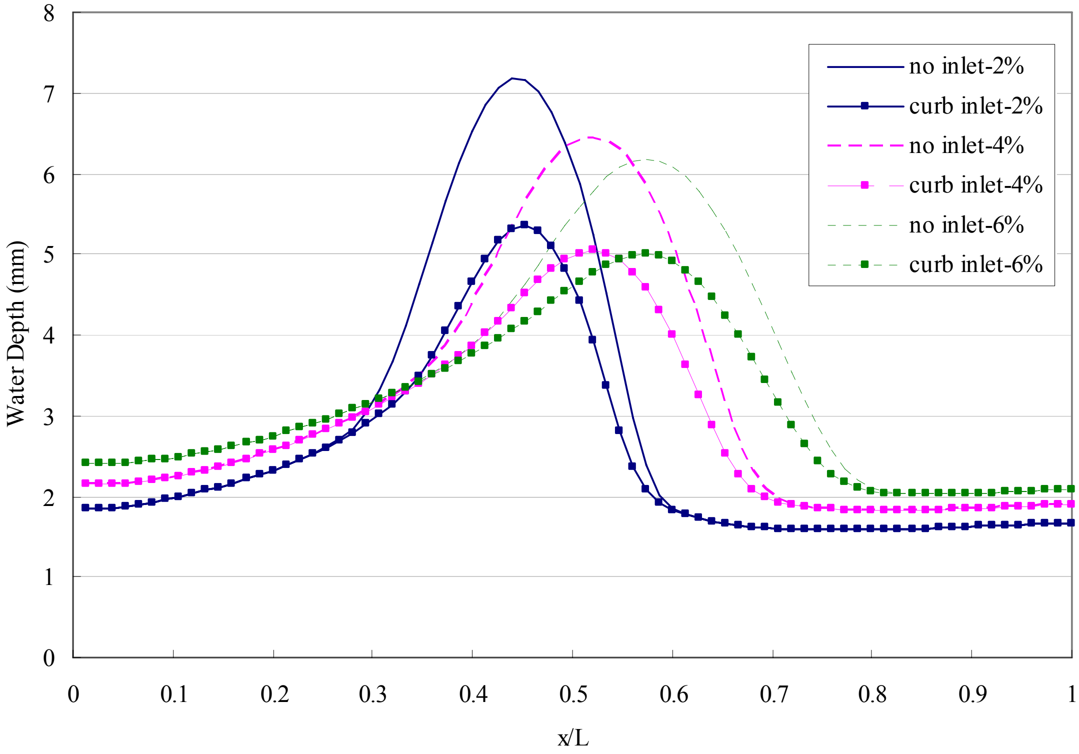


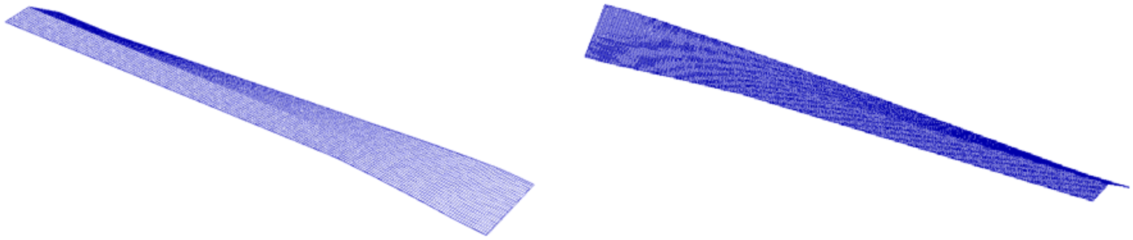
Figure 4.22: Performance of curb-opening inlet

The curb-opening inlet is further tested in Chapter 5 on various geometry and hydrologic conditions to investigate the optimal placement of the inlets that minimizes the flow depth on the road surface.

Chapter 5: Numerical Experiment

5.1 DESCRIPTION

Road surface geometry and rainfall intensity are the major variables that affect the property of sheet flow. The sensitivity of sheet flow to these variables is investigated through a numerical experiment on various road sections with different geometric configurations and rainfall intensity.



Type-I: Transition from normal crown to superelevation

Type-II: Transition from superelevation to normal crown

Figure 5.1: Types of the roadway surfaces used in the numerical experiment.

Various shapes of asphalt concrete pavement (ACP) surfaces were designed with GEOPAK with the following variables: L=460m, R=610m, design speed=100 km/hr, maximum superelevation rate=4% (based on TxDOT design manual). The estimated time of concentration is less than three minutes in all cases, so the time series solution at t=3min is considered as steady state solution. Sensitivity of stormwater runoff is tested on 270 scenarios out of the following cases.

1. 2, 3, and 4 traffic lanes in one direction (i.e. 4, 6, and 8 lanes on both directions) with 3-meter roadside shoulder.
2. 15 longitudinal slopes 0.1%, 0.2%, 0.3%, … , 0.9%, 1.0%, 2.0%, …, 5.0%, 6%.

3. The patterns of sheet flow on the surface are categorized into two types based on the geometry of the road. As shown in Figure 5.1a, *Type-I* starts with the normal crown on the upstream end and finishes with full superelevation on the downstream end. *Type-II* starts with full superelevation and finishes with the normal crown (see Figure 5.1b).
4. Rainfall intensity of 100mm/hr, 150mm/hr, 200mm/hr, and 250mm/hr.

From the geometric point of view, *Type-I* and *Type-II* roads comprise a curved section, so the two sections face each other at full superelevation. In other words, *Type-I* is the entrance to a curvature and *Type-II* is the exit to a straight section from the curved section. Even though they can be treated as a whole section combined, their geometries induce different sheet flow behaviors, and they need to be analyzed separately.

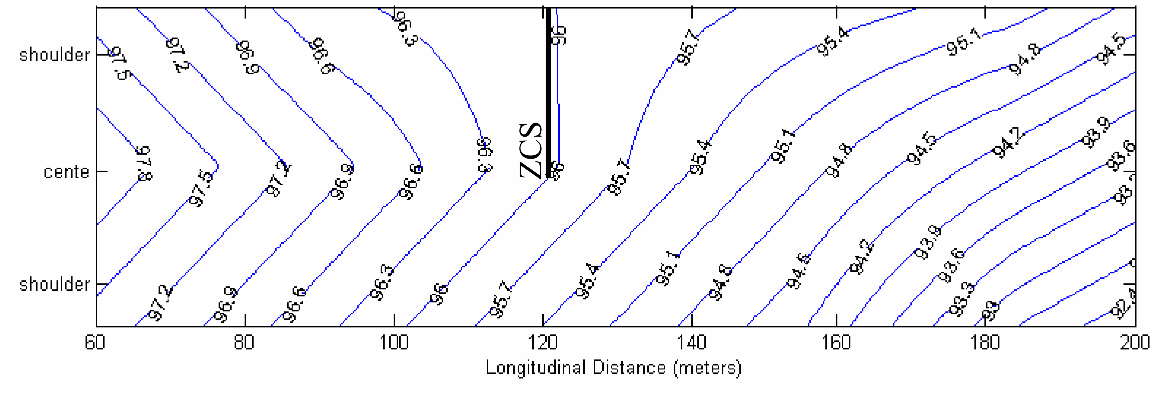


Figure 5.2: Contour plot of the surface elevation of a *Type-I* road (4-lane, S_x=1.0%, ZCS at 122m station, elevation in feet).

The cross slope of *Type-1* road begins with normal crown and gradually changes to full superelevation toward the downstream end as shown in Figure 5.2. Due to the transition of cross slope from the negative normal crown to the positive full superelevation, there exists a zero cross slope (ZCS) section on the outside-lanes of the road. For the case shown in Figure 5.2, the ZCS section is located at 122m on the outside lanes.

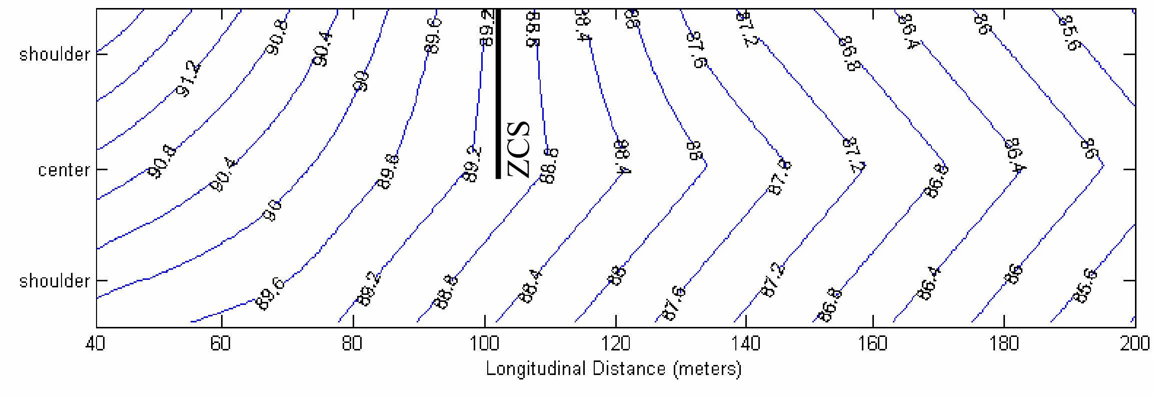


Figure 5.3: Contour plot of the surface elevation of a *Type-II* road (4-lane road, S_x=1.0%, ZCS at 103m station, elevation in feet).

The cross slope of *Type-II* road begins with full superelevation and gradually turns into normal crown as the roadway continues to a straight section at the downstream end as shown in Figure 5.3. The ZCS section exists on the outside lanes due to the transition in cross slope.

5.2 ANALYSIS

5.2.1 *Type-I*

“Mild” slope surfaces are defined as the surfaces on which the transition in the cross slope compromises the base slope, causing the longitudinal slope at the road side to be positive (uphill) or flat while the center line slope is negative (downhill). The profile of water depth on a mild slope surface is shown in Figure 5.4a labeled S_x=0.1%. On this surface, the cross slope changes from the normal crown (-2%) to a full superelevation (+3.8%). As one can see from the figure, there is an accumulation of the stormwater at the outside-lane edge near the location of zero cross slope (ZCS). Due to the positive longitudinal slope of the outside-edge, the stormwater drainage occurs toward the upstream direction (left direction in Figure 5.4) of the zero cross slope section (also see Figure 5.4a).

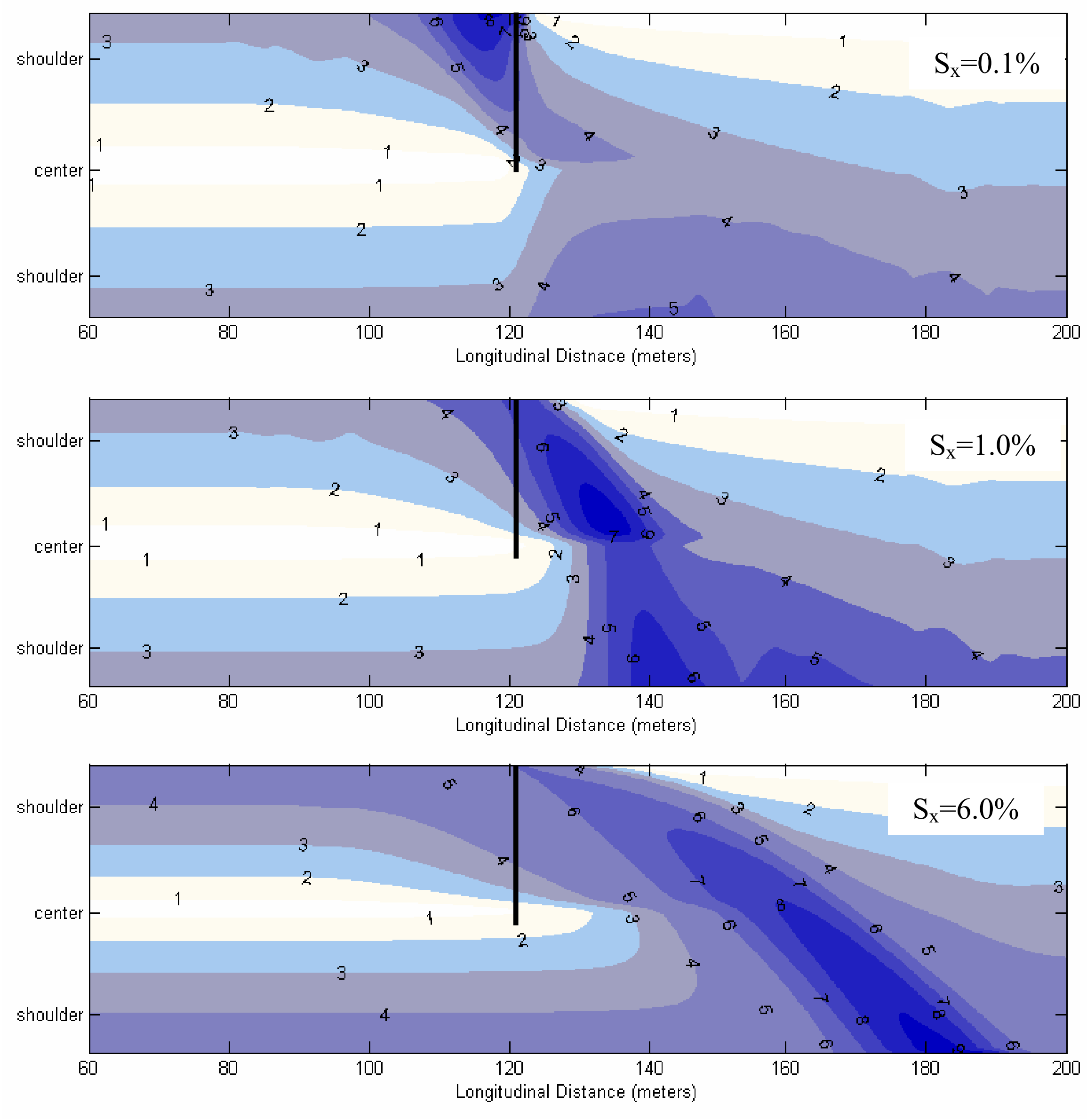


Figure 5.4: The profile of water depth at the steady state condition (*Type-I*, r=250mm/hr, 4-lane road, depth in millimeters).

For larger longitudinal slopes, the accumulated water at the outside-edge tends to flow inward towards the center of the road rather than draining to road side. On medium longitudinal slopes such as 1.0%, the sheet flow drains over the center of the road near the ZCS section. The flow forms a pond as it flows over the center of road near the ZCS section because of the sharp edge of the road center. On steep longitudinal slopes (e.g. 6.0%) the flow path becomes longer and the flow passes the road center far downstream from the ZCS section as steep longitudinal slope dictates the overall surface slope (see Figures 5.4c, 5.5c). The accumulated flow on these surfaces is typically continuous in profile with gradually increasing depth and size. Therefore, the maximum depth and flow rate occur at the downstream end (e.g. near 180m at the inside-lane edge in Figure 5.5c)

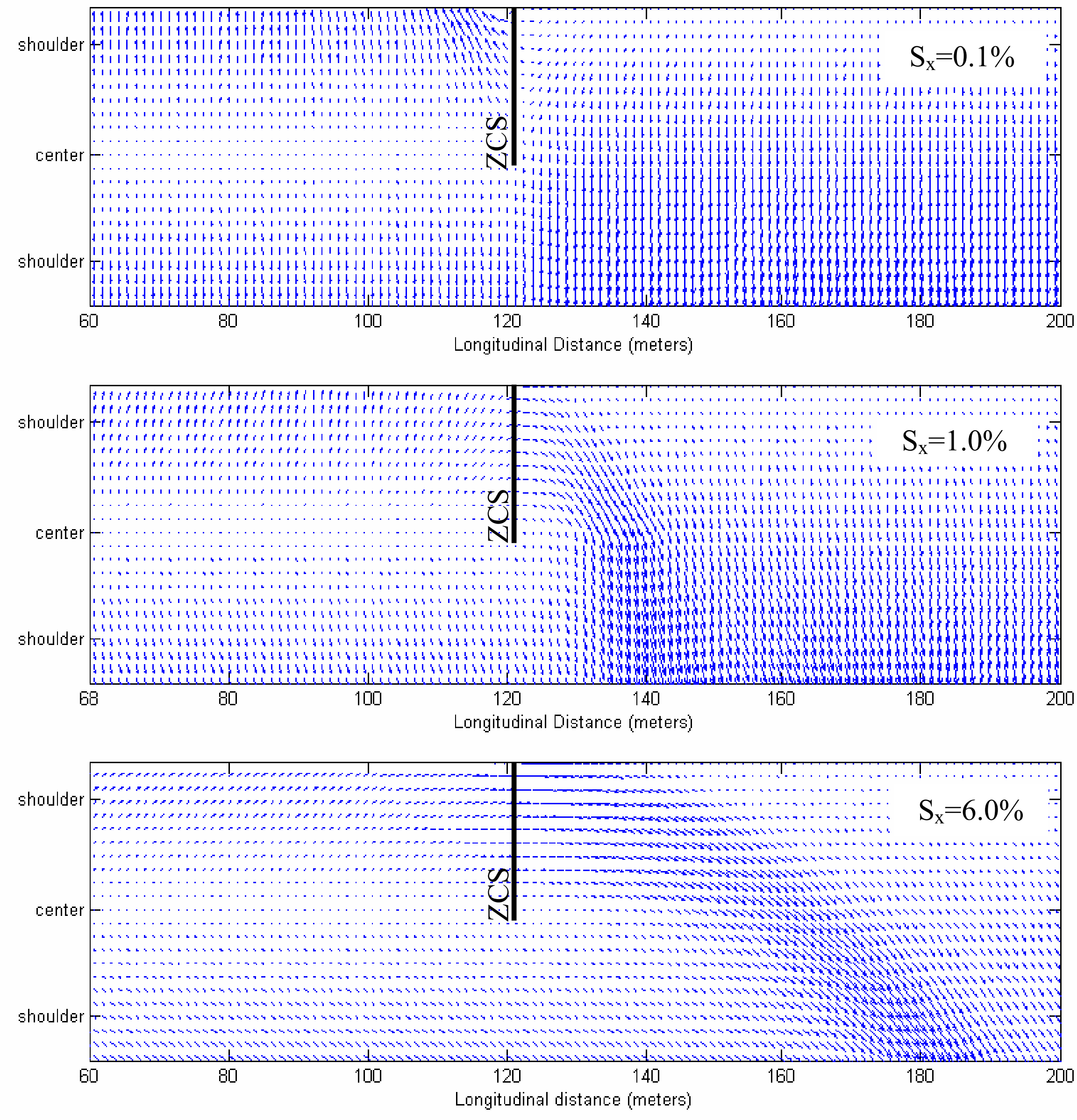


Figure 5.5: Vector plots of the unit flow rate at the steady state condition (*Type-I*, r=250mm/hr, 4-lane road, depth in millimeters).

The different patterns of sheet flow on *Type-I* surfaces are mostly related to base longitudinal slopes. This will be discussed later in the sensitivity analysis chapter in detail.

5.2.2 *Type-II*

The cross slope of *Type-II* roads changes from full superelevation to normal crown. As the cross slope of the outer half of the road changes from positive (full superelevation) to negative (normal crown), the stormwater runoff flowing inward (toward the road center) on the upstream gradually turns outward along the direction of the steepest slope. As shown in Figure 5.6, the tail of the concentrated flow grows larger and longer, as the longitudinal slope increases.

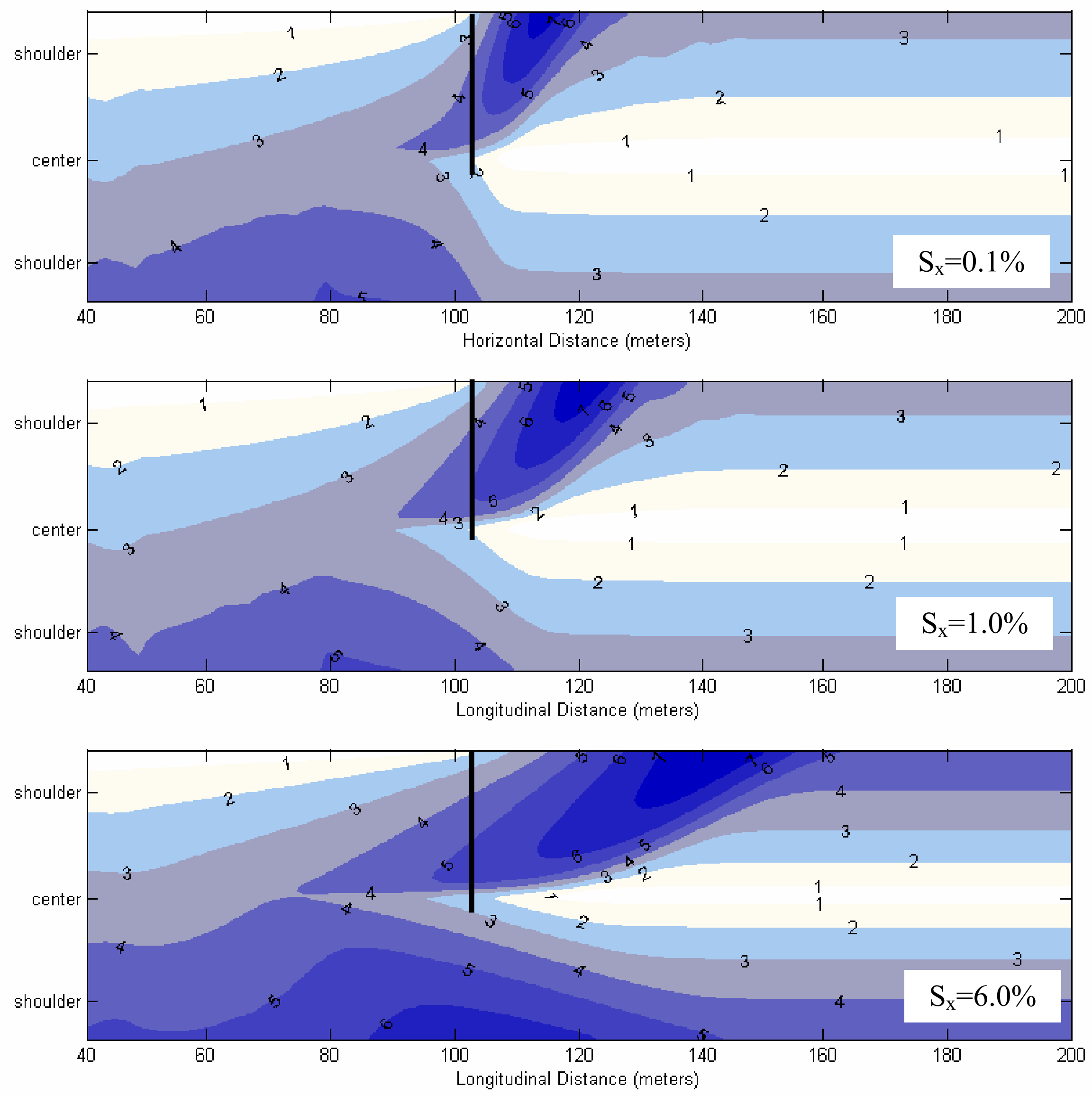


Figure 5.6: The profile of water depth at the steady state condition (*Type-II*, r=250mm/hr, 4-lane road, depth in millimeter).

Vector plot of unit flow (see Figure 5.7) shows the direction of sheet flow near the ZCS section on *Type-II* surfaces. In all cases the flow direction on the outside lanes is parallel to the traffic direction at the ZCS as it transition from flowing inward to outward, but the concentrated flow reaches the outside-lane edge farther downstream on greater longitudinal slopes.

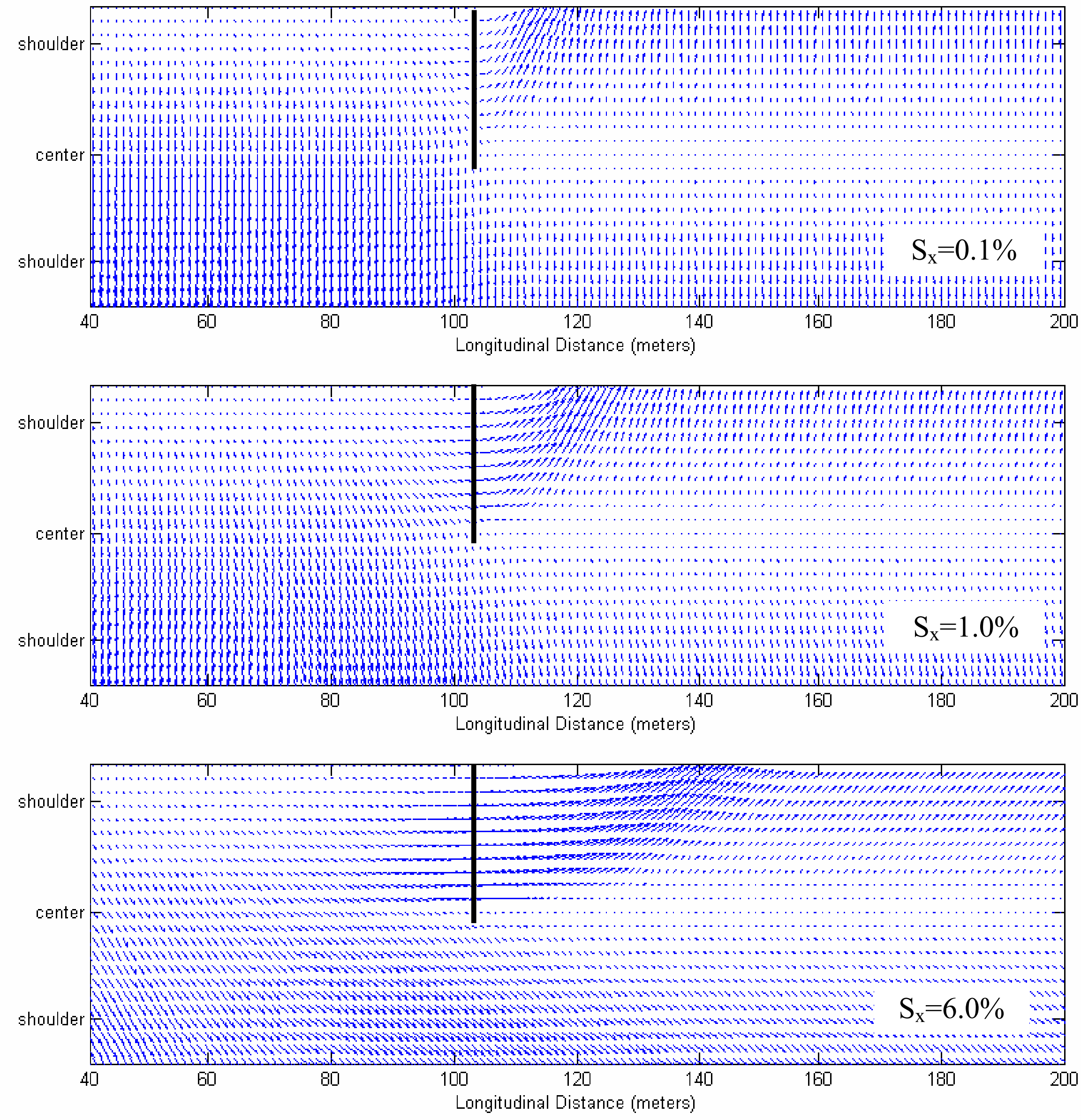


Figure 5.7: Vector plots of the unit flow rate at the steady state condition (*Type-II*, r=250mm/hr, 4-lane road).

The numerical test shows that the maximum depth is barely affected by longitudinal slope. On *Type-II* surfaces with longitudinal slopes varying from 0.1% to 6.0%, the estimated maximum depths were almost the same. This implies that the tendency of "spread" due to lateral gradient of water surface compromises the tendency of "accumulation" on these surfaces. Accordingly, the variation in the maximum depth on the *Type-II* surfaces in the numerical experiment is limited to less than one millimeter while the longitudinal slope varies from 0.1% to 6.0% as shown in Figure 5.8. Meanwhile, the size (width) and length of the accumulated flow increase as the longitudinal slope gets steeper.

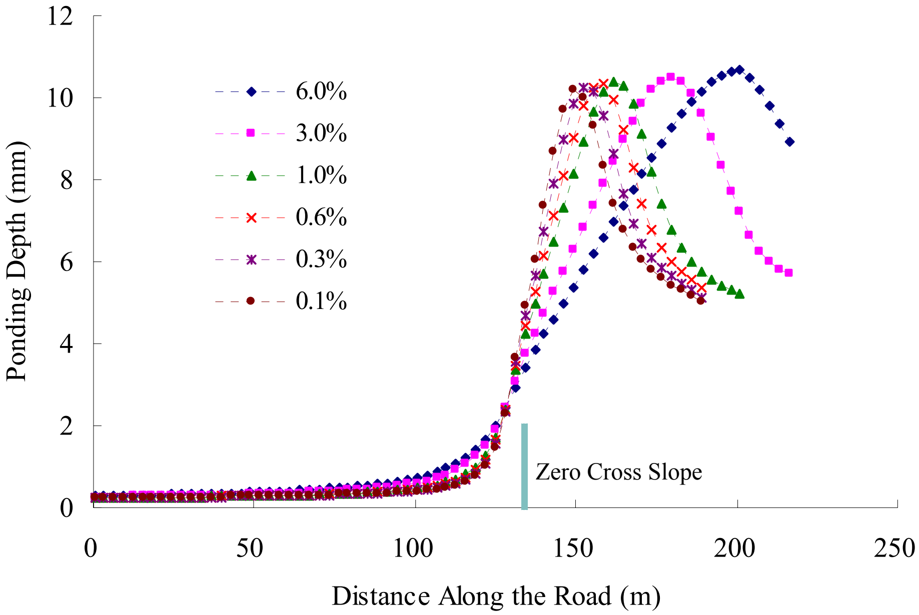


Figure 5.8: Longitudinal profile of ponding depth at the inside end of 8-lane road under 250mm/hr rainfall (*Type-II* roads).

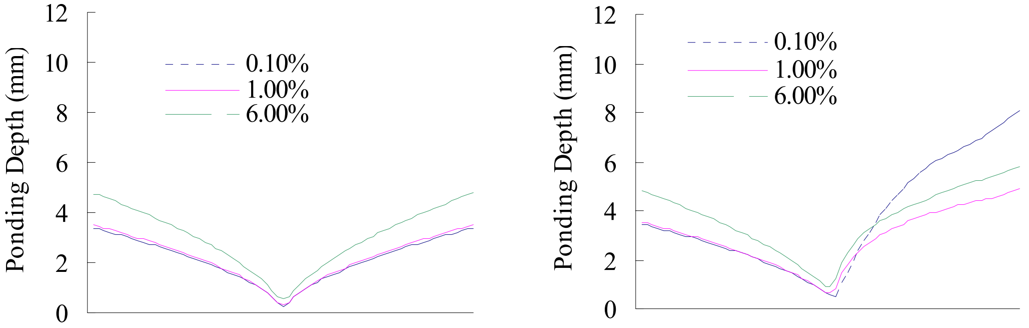
The area on the traffic lanes flooded by a certain depth or greater is much more on steep slope roads. In Figure 5.8, the distance along the road covered with 9mm or greater depth is approximately 10m on the 0.1% road. On the 6.0% road, much more distance (~35 meters) is flooded with the same depth of water. This implies that there tends to be more water ponded on the road surface on longitudinally steep roads. In conclusion, a near-flat longitudinal slope is recommended as the optimal longitudinal slope for *Type-II* roads on condition that it does not deteriorate the drainage along the roadside in case roadside curbs exist.

5.3 SENSITIVITY TEST

5.3.1 Longitudinal Slope

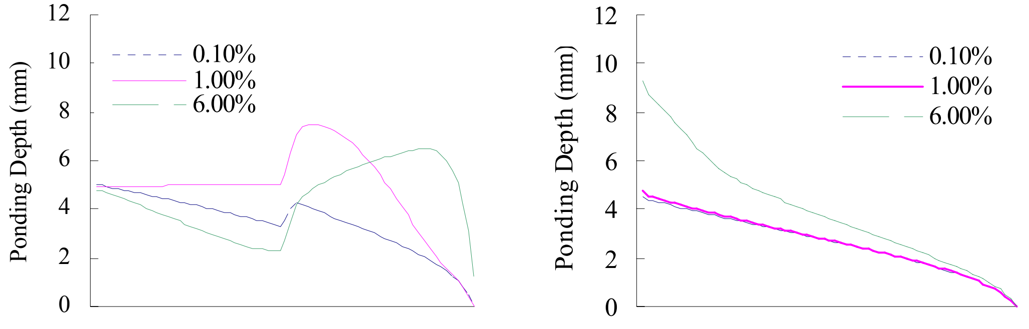
In this section, the degree of storm water accumulation and spread based on the level of longitudinal slope is investigated. Different longitudinal slopes affect the direction of the downhill slope of road surfaces and pattern of the flow. As mentioned in section 5.2, model simulation on *Type-I* surfaces shows interesting results when different longitudinal slopes are used. The accumulation of sheet flow on the 0.1% slope is limited to the outside-lane edge near the ZCS section. The positive (uphill) longitudinal slope toward downstream at the outside-lane edge generates reverse flow in this area. Since the flow paths head to road side edge, the accumulated flow does not propagate to the inside lanes. On 1.0% slope, there exists no positive longitudinal slope within the domain because the base longitudinal slope is larger than the compromising uphill slope generated by the superelevation transition. Therefore, the accumulated water at the outside-lane edge flows inward as it passes the ZCS section. The flow ponds near the center of road and spreads out as it flows over the center line.

Flow responses on different longitudinal slopes, 0.1%, 1.0%, and 6.0%, are directly compared at several locations as shown in Figure 5.9. The sections are at different longitudinal stations. The selected locations include the normal crown and the stations where the maximum depth occurs on each longitudinal slope: 0.1% road at 118m, 1.0% at 135m, and 6.0% at 185m. At normal crown, the profile of water depth on 0.1% road is slightly lower than 1.0% road, but there is little difference between the two. The profile on 6.0% road appears apparently higher than the other slopes. Assuming that the water flows along the steepest slope, the flow lengths on a 7.3m width (from the road center to side edge) road is 7.388m, 8.138m, and 23.10m for 0.1%, 1.0%, and 6.0 % roads, respectively. Flow depth is proportional to the length of flow, but is inversely proportional to surface slope. Considering the relative significance of flow length and surface slope, one can expect the extent of the differences in the water depth profiles at normal crown shown in Figure 5.9a.



(a) Cross section at normal crown

(b) Cross section at 118m



(c) Cross section at 134m

(d) Cross section at 185m

Figure 5.9: Cross sectional profile of water depth at different locations of the *Type-I* roads shown in Figure 5.4.

The 1.0% road at x=134m has a fairly uniform depth of 5mm on all the inside lanes, and the peak depth occurs on the outside lanes; therefore, almost three quarters of the cross section is flooded with at least 5mm depth of water. Water depth on the outside-lane edge becomes nearly zero as the cross slope turns positive (uphill) toward the outside edge after passing the ZCS, and the accumulation of water moves to inside lanes.

The longitudinal slope of the road affects the dynamics of the flow. As shown in Figure 5.10, the location of the maximum depth moves downstream as the base longitudinal slope of the road increases.

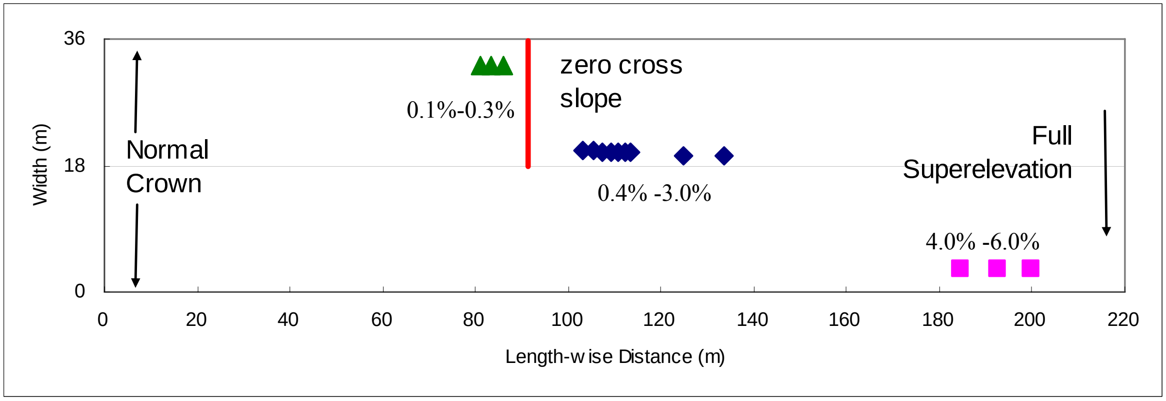


Figure 5.10: Locations where peak depth at the steady state condition occur on various longitudinal slope surfaces (*Type-I* roads).

On mild slope surfaces (S_x=0.1%-0.3%), the maximum depths are located at the outside-lane edge near the zero cross slope section (triangles). However, the location of the maximum depth jumps to the center of road as the longitudinal slope increases to S_x=0.4%-3.0% (diamonds). For longitudinal slope larger than 4.0%, the location of the maximum depth occurs at the inside edge far downstream from the zero cross slope section (squares). The relative change in the longitudinal slope at the outside-lane edge and the abrupt change in the cross slope at the center of may create such discontinuities. On mild longitudinal slopes, the increase in surface elevation due to the transition in cross slope overcomes the base longitudinal slope. As a result, the longitudinal slope of the outside-lane edge becomes positive while the center line and inside-lane edge have negative slopes. Because of these opposite slopes, there exists a stagnation point on the road, where the road surface forms a plateau, and the surface shape is similar to a saddle. A stagnation point may also be called a saddle point. Figure 5.11 shows an example of a stagnation point on a 0.1% slope road. Due to hyperbolic shape of the surface near the stagnation point, the flow diverges near the stagnation point to two opposite directions.

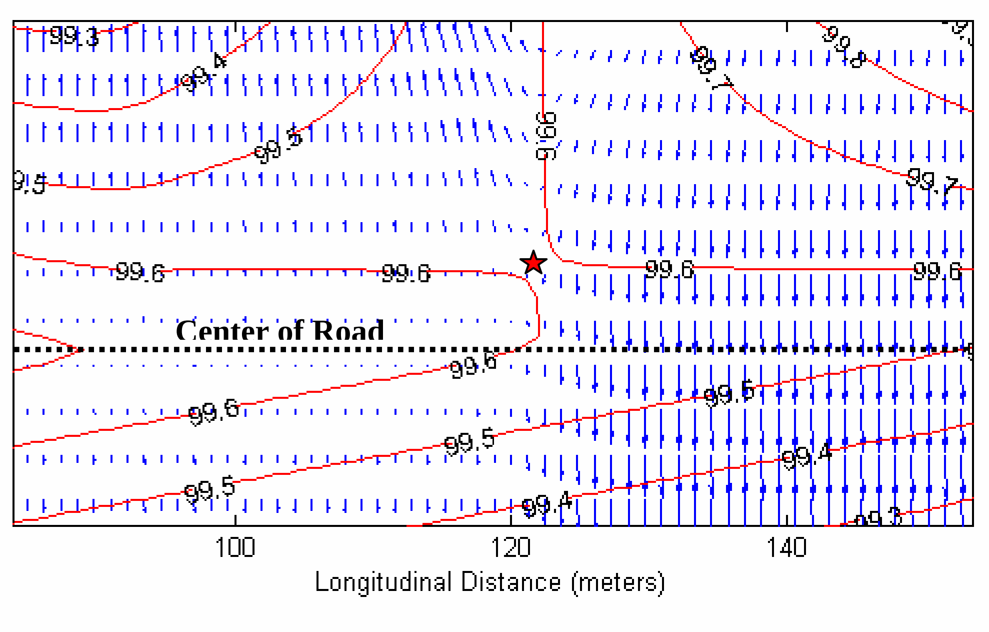


Figure 5.11: Saddle point at the ZCS section on a 0.1% slope road. Contours show surface elevation and arrows represent the direction and magnitude of unit flow on the surface at steady state.

The location of the stagnation point is affected by longitudinal slope. If the base longitudinal slope is small, the stagnation point is located near the center of road at the

ZCS section as shown in Figure 5.11. On larger longitudinal slopes, the location of the stagnation point moves to the outside-lane edge, and if the base longitudinal slope is large enough to make the slope of the outside-lane edge remain negative, it eventually disappears (see Figure 5.12). The gap between the locations of the peak water depths on 0.3% and 0.4% roads shown in Figure 5.10 can now be explained by this concept. The location of the stagnation point on the 0.4% must be close to the road side enough to make the diverged flow flowing toward the center larger than the flow going to the outside-lane edge.

The center of the road is the axis of rotation for normal crown shapes or superelevation transitions; therefore, the center is theoretically sharply angled in the transverse direction. As a result, the inside lanes on a superelevation transition section have steeper cross slope than outside lanes with the center of road sharply angled. Once the base longitudinal slope becomes large enough for the stagnation point to move off the road, the location of the maximum depth moves toward the center of road as water flows inward after passing the ZCS section. The accumulated water near the center of road spreads out quickly as it flows over the center line due to “weir” effect of the center of road.

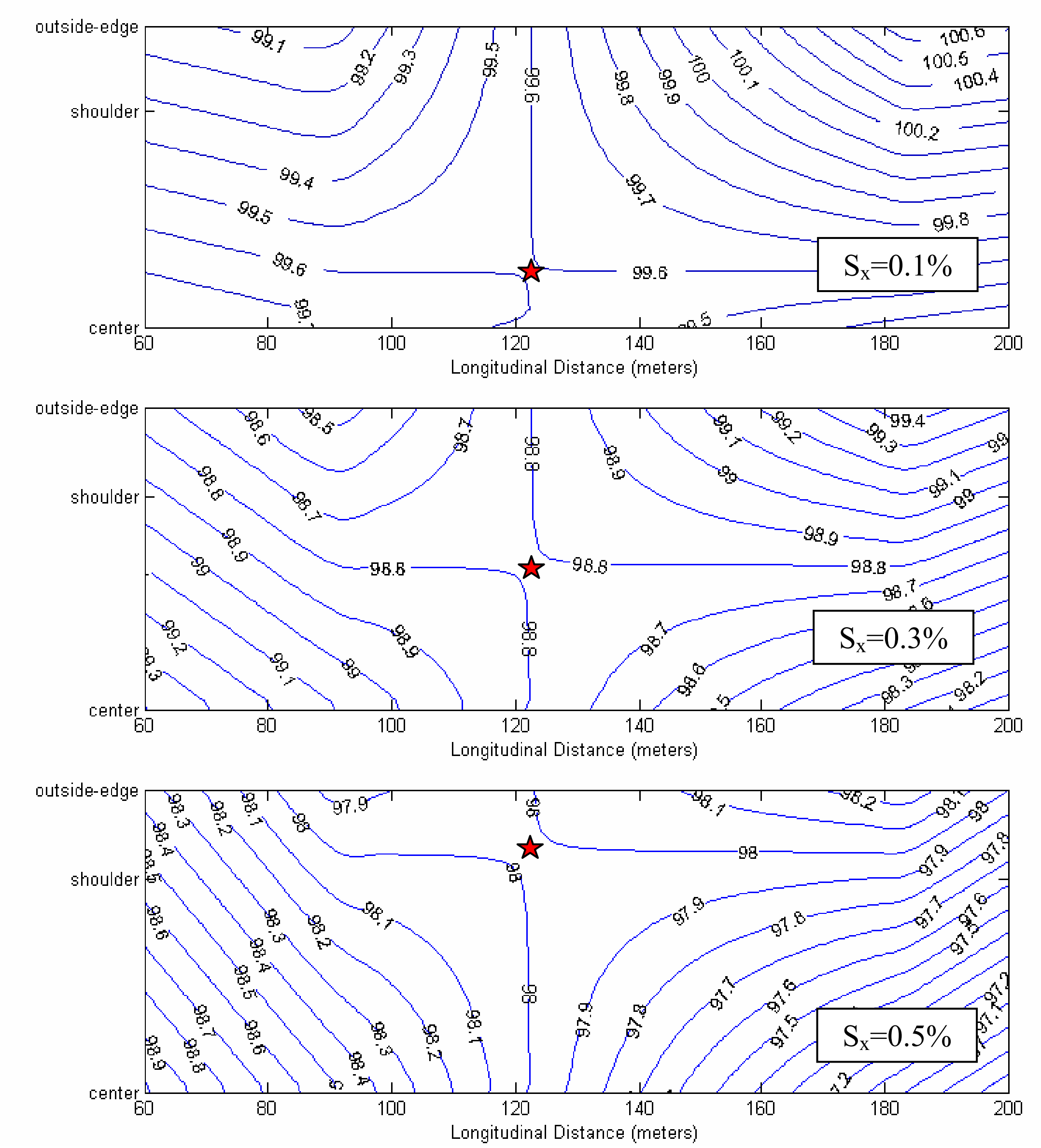


Figure 5.12: Contour of the surface elevation near the stagnation point (red star) on different slopes (*Type-I* roads).

On steep surfaces, the accumulated water flows far downstream from the ZCS section, where the shape of the center of road becomes much smoother as the cross slope turns to full superelevation. Therefore, there is little "weir" effect at the road center and the accumulated flow keeps its shape extending to the edge of inside-lane.

5.3.2 Rainfall Intensity

Sensitivity of the maximum depth on rainfall intensity is investigated. The maximum depths on different widths and longitudinal slopes under the rainfall intensity of 150mm/hr, 200mm/hr, and 250mm/hr are compared. Result shows that the maximum ponding depth is fairly linear with respect to different rainfall intensity values with $R^2 > 0.99$. However, the slope and intercept of linear regression vary with respect to the number of lanes and longitudinal slope. Nevertheless, with these linear relationships, one can simply interpolate the maximum depth on a superelevation transition section for any desired rainfall intensity. The slope and intercept values are summarized in Table 5.1. Figure 5.13 shows an example of the maximum ponding depth as a function of rainfall intensity on 4-, 6-, and 8-lane roads with S_x=1.0%.

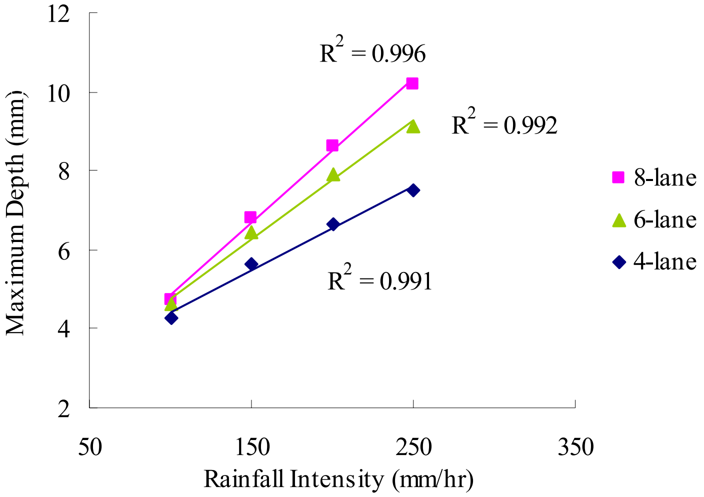


Figure 5.13: Linearity in the maximum ponding depth with respect to rainfall intensity (S_x=1.0%).

Table 5.1: Variables for linear regression of the maximum water depth w.r.t. rainfall intensity

Type	Sx (%)	4-lane road			6-lane road			8-lane road		
		slope	intercept	R^2	slope	intercept	R^2	slope	intercept	R^2
I	0.1	0.019	1.945	0.997	0.025	2.334	0.999	0.031	2.242	0.999
	0.3	0.018	1.724	1.000	0.024	1.972	0.999	0.029	1.913	0.999
	0.6	0.019	2.279	0.994	0.023	2.771	0.998	0.029	2.517	0.998
	1.0	0.021	2.269	0.991	0.030	1.801	0.992	0.036	1.231	0.996
	3.0	0.024	2.089	0.991	0.031	1.941	0.995	0.039	0.823	0.999
	6.0	0.030	1.587	0.985	0.038	0.840	0.999	0.041	0.631	1.000
II	0.1	0.018	2.222	0.993	0.023	2.732	0.998	0.029	2.445	0.998
	0.3	0.020	2.249	0.990	0.027	2.298	0.994	0.037	0.948	0.998
	0.6	0.019	2.254	0.990	0.025	2.559	0.996	0.033	1.825	0.998
	1.0	0.019	2.257	0.990	0.026	2.483	0.996	0.034	1.551	0.998
	3.0	0.020	2.249	0.990	0.027	2.298	0.994	0.037	0.948	0.998
	6.0	0.021	2.215	0.989	0.029	2.099	0.993	0.039	0.619	0.999

($h_{max}=ar+b$, *r: rainfall intensity [mm/hr],* h_{max}*: maximum water depth [mm], a: slope, b: intercept*)

5.3.3 Surface Roughness

Manning’s equation is an empirical flow equation which has been popularly used among civil engineers for decades. In this equation, flow velocity and depth are related to two geometric variables: bed roughness and friction slope. The Manning’s *n* value represents the roughness of bed surface. Standard Manning’s coefficient values for corresponding bed surface material are listed in Table 5.1.

Table 5.2: Manning’s n for Street and Pavement Gutters (HEC-22, 2001)

Type of Gutter or Pavement	Manning’s n
Concrete Gutter, troweled finish	0.012
Asphalt Pavement:	
Smooth texture	0.013
Rough texture	0.016
Concrete gutter-asphalt pavement:	
Smooth	0.013
Rough	0.015
Concrete pavement:	
Float finish	0.014
Broom finish	0.016
For gutters with small slope, where sediment may accumulate, increase above values of "n" by	0.02
Reference: USDOT, FHWA, HDS-3(36)	

As stated in Chapter 1, the experimental result shows that the Manning's coefficient is not a function of the flow on the surface but a function of surface roughness in terms of mean grain size.

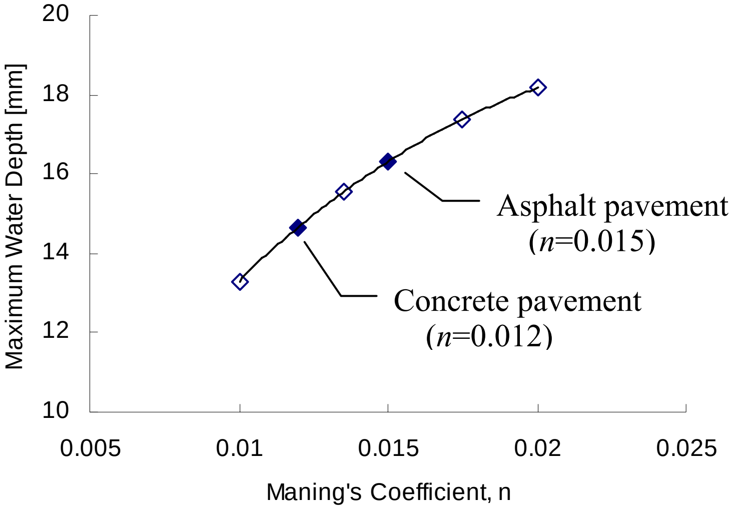
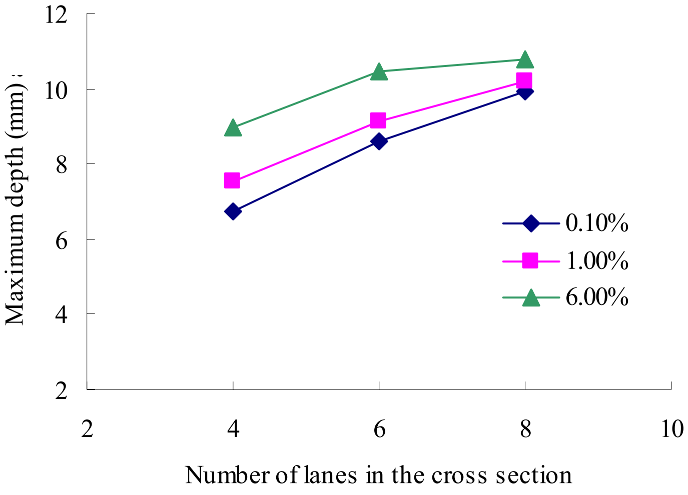


Figure 5.14: Flow responses on the downstream end of a 100 feet road surface with Sx=0.1% for different values of surface roughness coefficient.

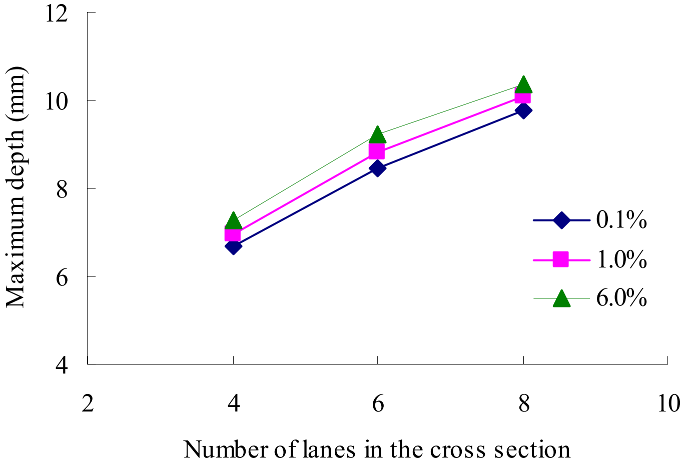
When storm water drains from the road surface, the friction between water and road surface drags retards the flow. This drag force is normally proportional to the surface roughness, and water depth is larger on rougher surfaces. The model result confirms this as shown in Figure 5.14. The maximum water depth at steady state becomes larger as the Manning's *n* increases. For the same geometry and rainfall intensity, the asphalt pavement surface (n=0.015) has 10% more water (depth) than the concrete pavement surfaces (n=0.012).

5.3.4 Number of Traffic Lanes

According to the AASHTO Green Book (2004), the length of superelevation runoff is proportional to the number of lanes on either side of the center. The length of stormwater runoff on the road surface tends to increase as the number of lanes increases. The sensitivity of the maximum depth to the number of lanes can not be directly measured because the increase of number of lanes is equivalent to increase of drainage area. Therefore, one should be careful when interpreting the result such as those shown in Figures 5.15 and 5.16 because the roads have different superelevation runoff lengths and drainage areas.



(a) *Type-I* roads

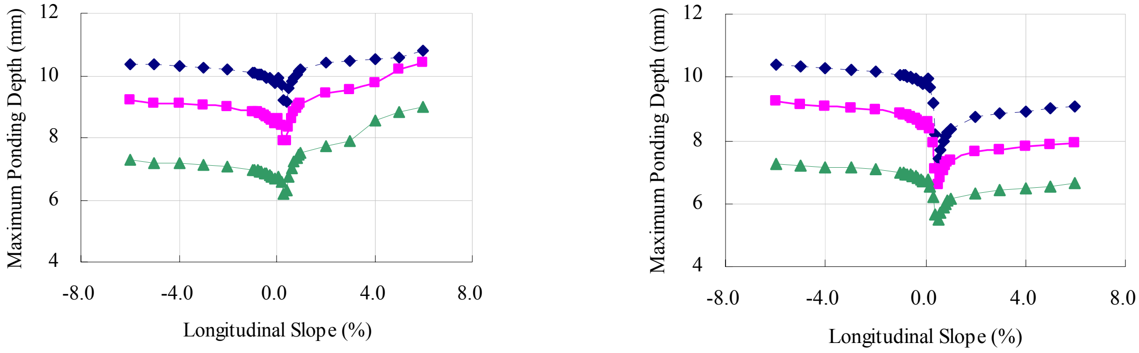


(b) *Type-II* roads

Figure 5.15: Maximum ponding depths on the traffic lanes (r=250mm/hr).

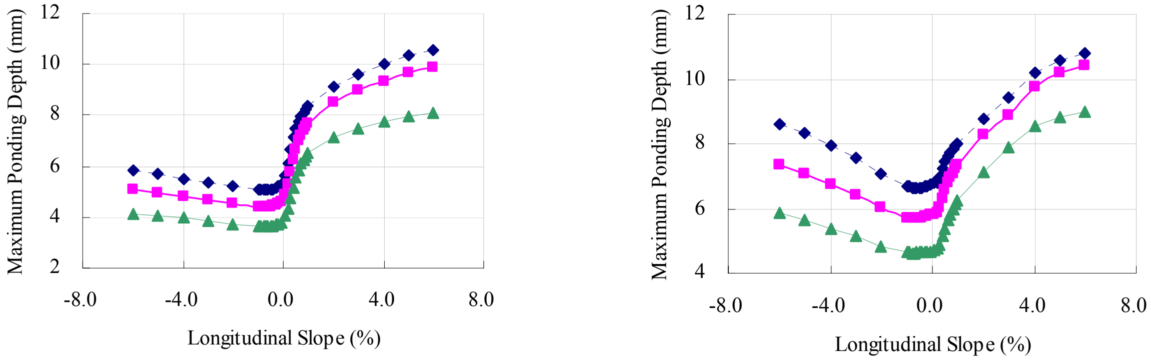
As expected, the roads with larger width and steeper longitudinal slope have more ponding depths than the others. On *Type I* roads, the impact of longitudinal slope is significant on the 4-lane road, but the gap becomes narrower as the number of lanes increases. As discussed in Section 5.3.1, the location of the maximum depth and flow pattern vary with respect to longitudinal slope; thus, converging trend on *Type-I* roads shown in Figure 5.15(a) is considered trivial.

On both road types, the maximum depth shows a trend of curvature with respect to number of lanes. This can be explained theoretically by kinematic wave theory in which water depth is proportional to flow length in a nonlinear style ($\sim L^{3/5}$). It is very likely that the overall trend of maximum depth over number of lanes comes from this relationship. Figure 5.16 gives another perspective for the analysis: the maximum depths on the entire traffic lanes, inside-lane edge, outside-lane edge, and the center of the roads with different widths are presented. In these figures, positive and negative values of longitudinal slope denote the *Type-I* and *Type-II* roads, respectively.



(a) Entire traffic lanes

(b) Outside-edge of traffic lane



(c) Center of road

(d) Inside-edge of traffic lane

Figure 5.16: Maximum ponding depth (r=250mm/hr).

It is interesting that the profile of maximum depth at the center of road is similar to that of the inside-edge of lanes. The curved profile observed in Figure 5.16 consistently repeats in this figure. The gap in the profile between 8-lane and 6-lane roads is overall less than that between 6-lane and 4-lane roads, and is especially significant at the center and inside-edge of *Type-I* road.

5.3.5 Location of Curb-opening Inlets

A curb-opening inlet is often used to collect gutter flow on the highway pavement when roadside curbs exist. HEC-22 (FHWA, 1996) recommends placing an inlet immediately upstream of the location of zero cross slope. In this chapter, flow response with respect to the location of a curb-opening inlet is investigated. A 10-foot long curb-opening inlet is placed on the outside-lane edge at the location of zero cross slope and every two feet upstream from the ZCS section (schematic drawing is presented in Figure 4.22). Result is presented in Figure 5.17 where horizontal axis represents the distance of the downstream-side end of the inlet from the ZCS section and vertical axis denotes the maximum ponding depth within the domain (entire traffic lanes of a superelevation transition section). As shown in the figure, the location of inlet affects the maximum ponding depth within the domain: the maximum depth decreases as the location of the inlet moves toward upstream from ZCS section, then reaches a lower limit at 1.5m, and keeps increasing as the location of inlet is moved farther from the ZCS section.

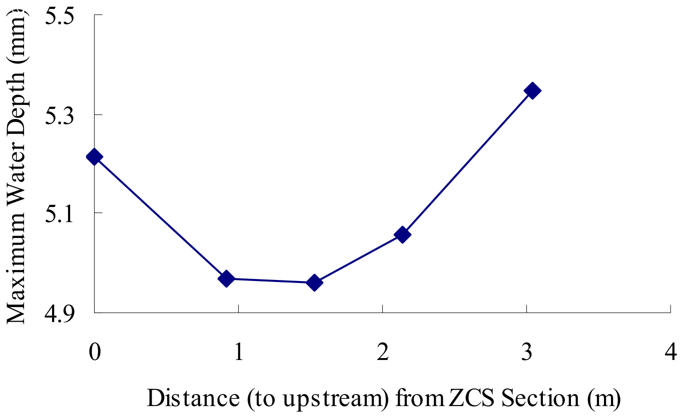


Figure 5.17: Maximum ponding depths on the traffic lanes (shoulder area excluded) on *Type-I* roads (r=250mm/hr).

The result shown in Figure 5.17 implies that the efficiency of inlet becomes highest with location 1.5m upstream from ZCS section because the gutter flow on the roadside starts to spread out before it reaches the ZCS section. Even though cross slope is still downhill to the outside-lane edge, the gradient of water surface promotes a lateral flow into traffic lanes. As the location of inlet is moved farther upstream, the inlet becomes inefficient because it fails to capture the flow coming from the main road accumulating between the inlet and ZCS section. However, we can not claim that the optimal location of inlet (1.5m in this case) does have significant meaning in terms of flow control on the road, for the difference in the maximum depth between the locations of an inlet at ZCS section and 5 feet away is less than 1 mm. More simulations with different sizes of inlets on various surfaces will help understanding the significance of curb-opening inlets. Moreover, grate inlets are another type of inlets popularly used in highways. An investigation for finding the optimal locations of these inlets or for comparing the efficiencies of different inlets is left as future work.

5.4 RESIDENCE TIME OF STORMWATER RUNOFF

Extended flow path results in the increase of residence time, T_r, of stormwater runoff near the location of ZCS on superelevation transitions. In this section, impact of superelevation transition is studied by estimating residence time of stormwater runoff near ZCS sections. Relative differences of the estimated T_r values for superelevation transitions and normal crown sections are compared. Stormwater runoff Residence time is estimated by T_r = Volume / Discharge. To make the calculation, a section of roadway that includes the ZCS station is selected. Then, using the steady-state depth in each cell, the cell storage volume is calculated from the cell depth and area. The volumues for cells within the roadway section are then added over the selected domain to get the total volume of water. The discharge is calculated using Discharge = ∑(Cell Area x Rainfall Intensity). The residence time for normal crown is calculated analytically using kinematic wave theory. Figure 5.18 shows the difference in residence time between a superelevation transition section and a normal crown section for the following variables: n = 0.015, r = 250 mm/hr, number of lanes is 4, 6, and 8, shoulder width = 3 m, and variable base longitudinal slopes. The difference in residence time shown in Figure 5.18 is purely due to the difference in geometry of the roads; there is approximately a thirty percent increase in T_r values associated with the greater water depth on the roadway.

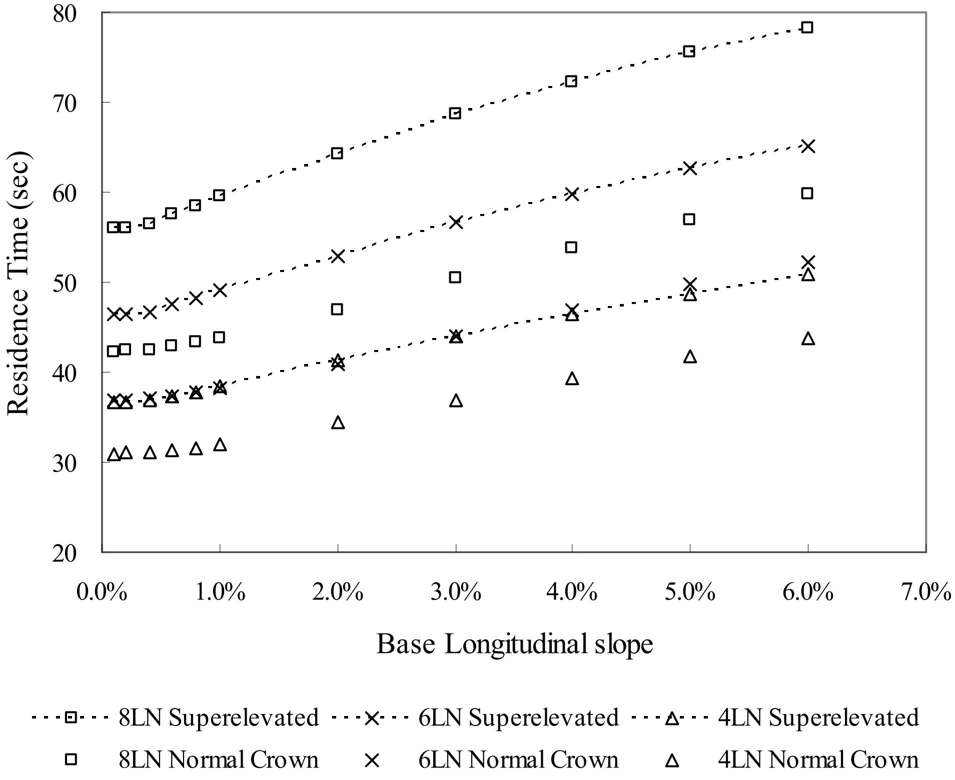


Figure 5.18: Residence time of stormwater runoff

Results show that the residence time increases as the base longitudinal slope increases. This implies that more water exists on the roadway on longitudinally steep grades than on flat sections. More importantly, residence time is always larger on superelevation transition sections compared with normal crown sections. The difference in residence time with respect to base longitudinal slope is fairly consistent; however, the magnitude of the difference as the number of traffic lanes increases. This may be observed more clearly in Figure 5.19, which shows the box-plot of the difference in Residence time (*T_r*) between superelevation transition and normal crown sections. The trend of the box-plot shows that the difference in residence time becomes larger as the number of traffic lanes increases. Furthermore, the variation of the difference increases as the number of lanes increases.

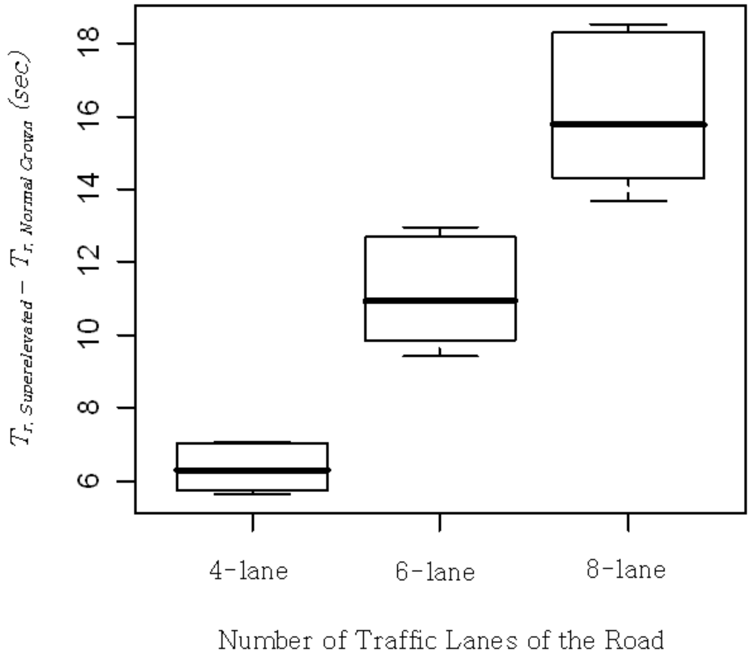


Figure 5.19: Box-plot of the difference in residence time between superelevation transition and normal crown sections

Impact of superelevation transition length on residence time is investigated. Table 5.3 shows the relative difference of residence time between normal crown and superelevation transition sections with S_x=0.02, n=0.015, r=250mm/hr. On the normal crown sections, residence time does not change with respect to the length of road, while it increases as the length of superelevation transition increases. The relative error is estimated 24.2% for 30.5m road to 34.5% for 91.4m road.

Table 5.3: Estimated difference in residence time of stormwater runoff between normal crown and superelevation transition sections

Length of Road		Residence Time (sec)		Error (%)
(ft)	(m)	Normal Crown	Superelevation	
100	30.5	28.13	37.10	24.2
200	61.0	28.13	41.73	32.6
300	91.4	28.13	42.94	34.5

5.5 DISCUSSION

If the maximum depth is the only concern, one may conclude that the optimal longitudinal slope is between 0.3% and 0.4% for *Type-I* roads (see Figure 5.16(a)). Meanwhile, very small difference of the maximum water depth is observed over different longitudinal slopes on *Type-II* roads (Figure 5.8). In most cases, a dominant portion of the traffic lanes on superelevation transition is covered with more than 4 mm of sheet flow. Huebner et al. (1986) suggested that vehicles traveling more than 45km/hr are at risk of hydroplaning when a road is flooded with 4 mm depth of water. The result of the model simulations shows that the probability of hydroplaning is high at superelevation transition areas in wet conditions. In addition, the existence of accumulated water across the road on steep roads may exert drag forces on traveling vehicles. For instance, it takes only two seconds for a vehicle traveling at 50 km/hr on a 0.1% slope road to experience the ponding depth increases from negligible amount to the peak depth (Δh=8mm) because of the steep longitudinal gradient of ponding depth (see Figure 5.8). In this circumstance, the driver has very little time to recognize the ponding and to reduce speed accordingly. The combination of centrifugal force, downhill slope, and the loss of friction due to hydroplaning can create an environment favorable to critical accidents on these specific locations of a highway. Furthermore, the differential ponding depth in the cross sectional direction near ZCS as shown in Figure 5.9 can create differential drag forces on each side of a vehicle causing a rotational torque, which initiates a rotational motion of the vehicle resulting in complete failure of control.

Chapter 6: Model Application

6.1 INTRODUCTION

RM 2338 stretches to North to meet TX 970 road from the city of Georgetown in Williamson County, Texas. It is a rural highway, designed for a maximum speed of 96 km/hr, with 4 traffic lanes (*w*=3.7m each), 3m shoulders at both sides, and a 4.3m median at the center. Whereas many rural highways are separated by vegetated median at the center, this road is undivided but has a paved median, which means both sides of the road are hydraulically connected (see Figure 6.2). Roadside edges are curb-less allowing lateral drainage of stormwater from the pavement.

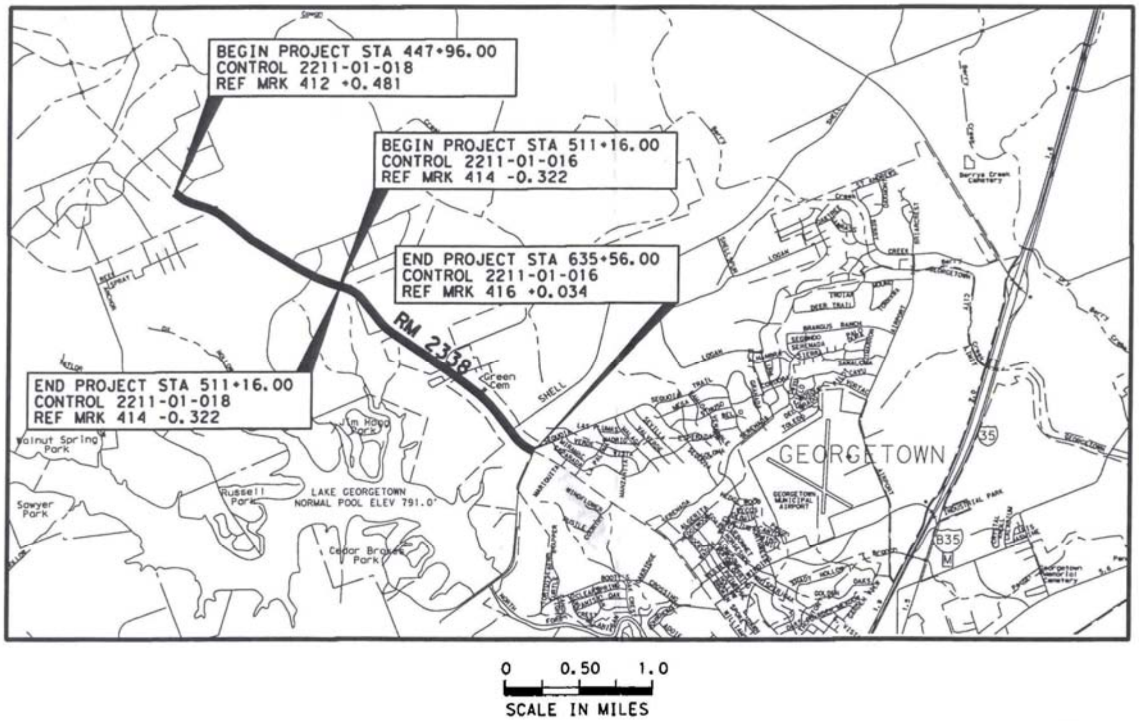
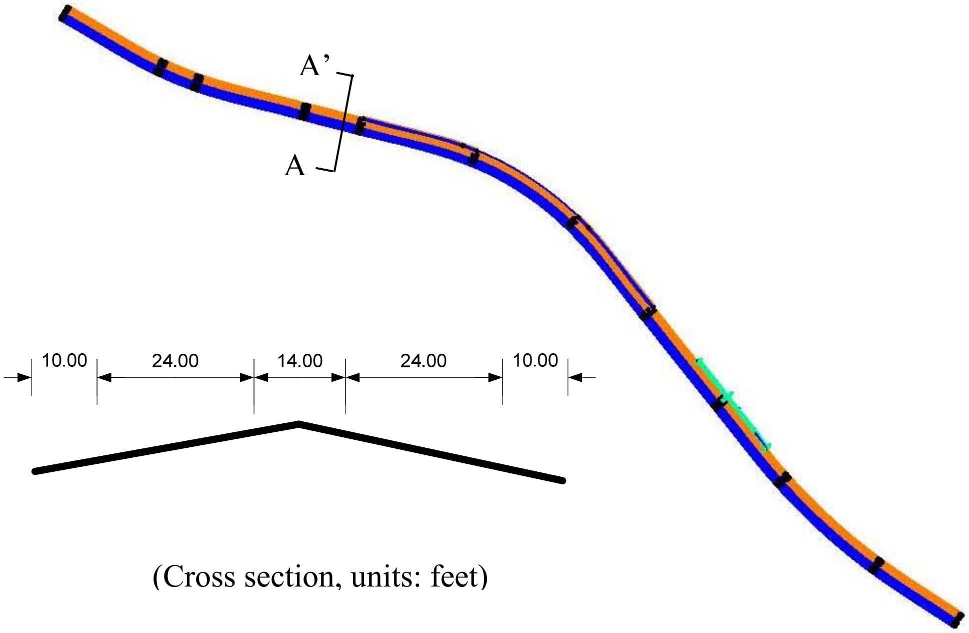


Figure 6.1: RM 2338 in Williamson County, Texas



(Cross section, units: feet)

Figure 6.2: Profile of RM 2338 in the GEOPAK shape file (Dimensions in feet)

This road is undergoing a renovation to improve stormwater drainage by adjusting vertical alignment, constructing ditches, and so forth. Since stormwater runoff near zero cross slope on superelevation transitions is of concern in this research, the developed numerical model is applied on several curved sections of this road. Four different superelevation transition areas were selected for the model application. Figure 6.2 shows the shape file of a 1524m section of the road. Using the DTM interface, the geometry of the road surface at 4 different superelevation transition areas was generated. The selected segments are named as #1, #2, #3, and #4: segments #1 and #3 are *Type-I* surfaces and segments #2 and #4 are *Type-II* surfaces. Texas Hydraulic Design Manual recommends using 25-year return period storm in designing hydraulic structures. In Williamson County, this is as much as 250mm/hr of rainfall intensity and this value is used for the model simulation. Details of the geometric properties and simulation result are presented in Table 1.

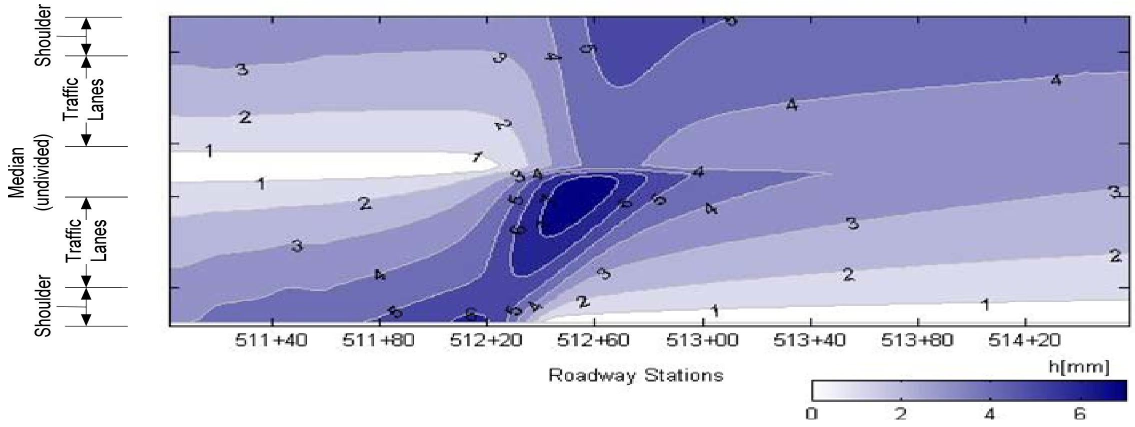
Table 6.1: Properties of the road segments and estimated values

Properties	Type-I		Type-II	
	SEG#1	SEG#3	SEG#2	SEG#4
Segment Type	IN[a]	IN[a]	OUT[b]	OUT[b]
Radius of Curvature	585m	875m	585m	875m
Transition Length	216m	142m	160m	142m
Superelevation rate	5.6%	4.8%	5.6%	4.8%
Longitudinal Slope	0.52%	0.24%	0.61%	1.28%
Domain Size	160m*25m	152m*25m	165m*25m	140m*25m
# of Grids (Nx, Ny)	(260, 103)	(250,103)	(270, 103)	(230,103)
Road Side B.C.	Open	Open	Open	Open
Max. Water Depth	7.5mm	7.5mm	8.2mm	8.4mm
Max. Unit Discharge [m2/s]	2.30E-03	2.30E-03	2.38E-03	2.61E-03

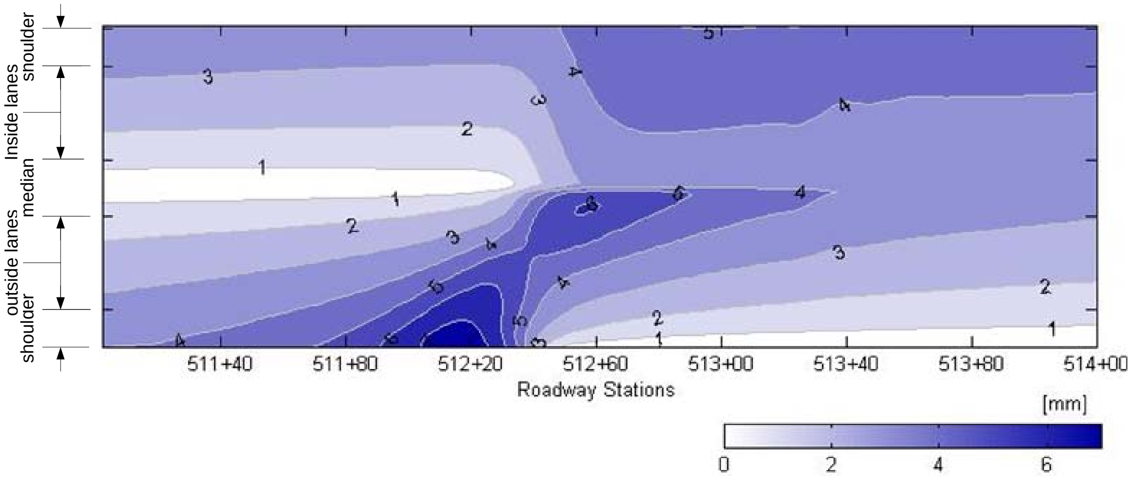
a: Normal crown on the upstream, and superelevated on the downstream
b: Superelevated on the upstream, and normal crown on the downstream

6.2 RESULT AND DISCUSSION

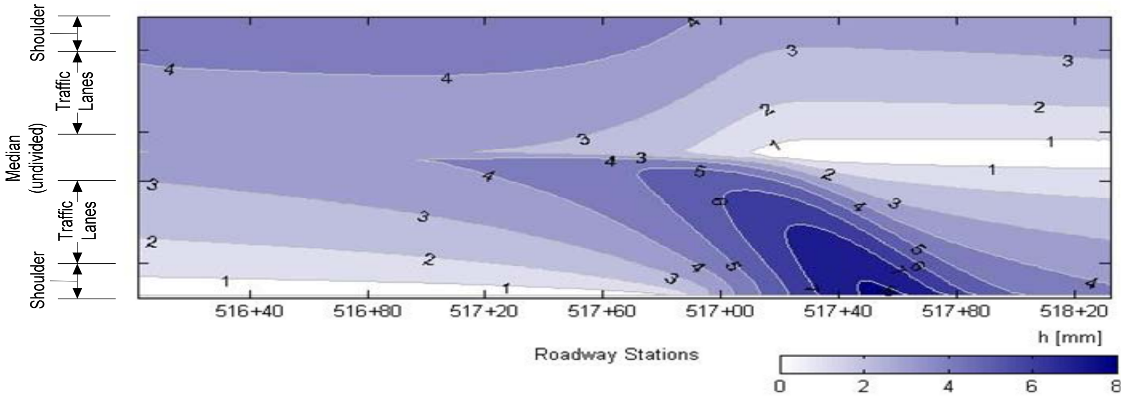
Based on the properties of the selected segments of the road, flow responses under the 25-year storm (250mm/hr) were simulated with the developed model. Overall trend of the simulated results is quite similar to that of the numerical experiment conducted in Chapter 5. Knowing that the base longitudinal slopes of segments #1 and #3 are 0.52% and 0.24%, respectively, one may find the result shown in Figure 6.3 is not very different from the result of the numerical experiment. The location of the maximum depth and the base longitudinal slope are consistent with the findings shown Figure 5.10. The longitudinal slope of segment #4 (S_x=1.28%) is steeper than that of segment #2 (S_x=0.61%), and segment #4 has longer and wider accumulated flow than that of segment #2.



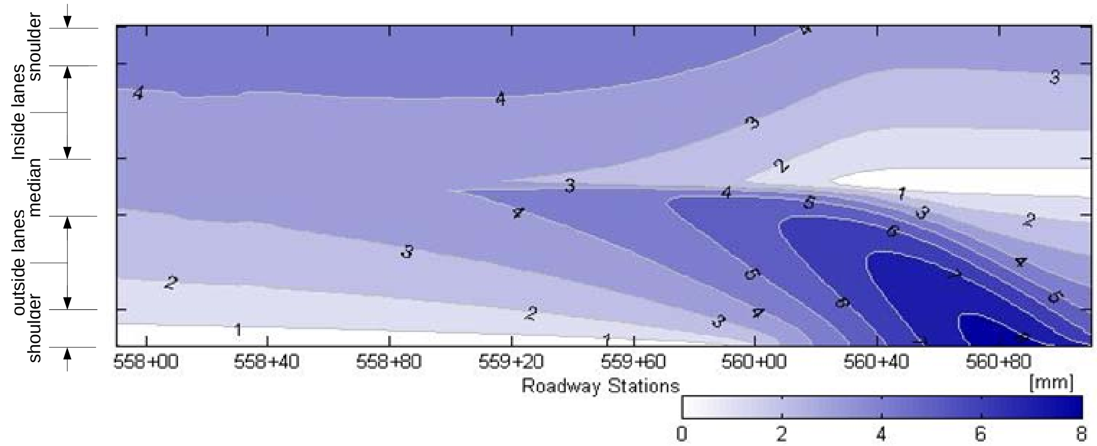
(a) Segment #1



(b) Segment #3



(c) Segment #2



(d) Segment #4

Figure 6.3: Profile of sheet flow on the road at steady state condition (r=250mm/hr, depth in millimeter).

All selected segments have local accumulation of sheet flow on the outside lanes under rainfall events on condition that the flow is not disturbed by traffic vehicles. This assumption becomes critical if the road is full of traffic during the rainfall; however, if the traffic is not heavy, it is often empty for several minutes, which is comparable to the time of concentration (typically 2 to 4 minutes), and the simulation results become very plausible. The first vehicle traveling toward the downhill direction on a superelevation transition area already under steady state (from left to right in Figure 6.3) may encounter a sudden increase in ponding depth within very short distance and time to cause hydroplaning.

Minimizing the spread of sheet flow on the road surface is one of the main concerns for highway designers. The distribution of ponding depth is presented in Figure 6.4 in a way that the ponding depths on all grid points are sorted and plotted with respect to the nondimensionalized rank of data. Similar graphs are found among flow duration curves in hydrology where the time is ranked in the horizontal axis rather than space.

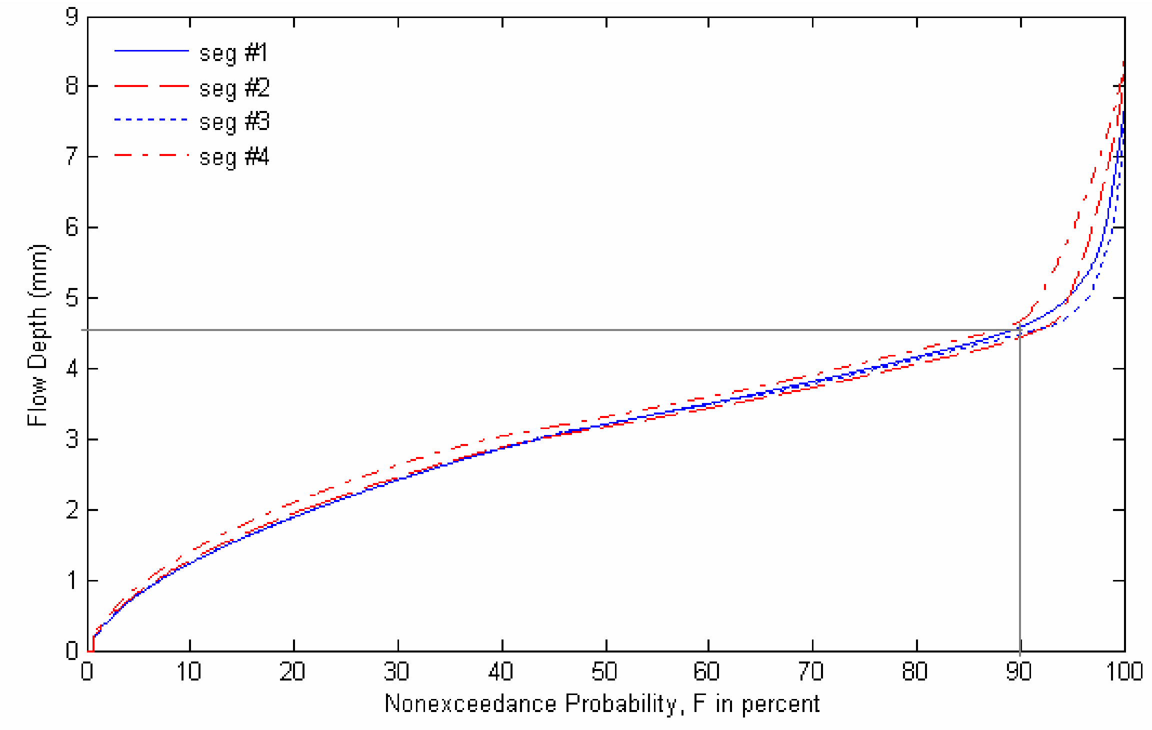


Figure 6.4: Nonexceedance probabilistic distributions of ponding depth for selected 4 segments

The area under the distribution curves in Figure 6.4 is equal to the average flow depth for the corresponding area of the surface. On the upper end, approximately 10 percent of the domain is flooded by 4.5mm or deeper depth of water at the steady state. The steep slope on upper end implies that water depth of 4.5mm or deeper occurs in small areas, which may encourage traffic vehicles to hydroplane near these areas. On the other hand, mild and fairly linear slope for the other 90% of domain implies that flow depth varies gradually over space in most of the domain. In conclusion, the nonexceedance probabilistic distribution curve re-confirms that the flow accumulation occurs in superelevation transition areas as spatially isolated events. The trends of flow accumulation are very similar in all segments, while the slopes of profile at the upper end on *Type-I* roads (seg#1 and seg#3) are shaper in smaller areas.

The flow responses on the selected segments in RM2338 may be compared to the results of the comprehensive numerical experiment conducted in the previous chapter. Therefore, one may use results of the numerical experiment to further analyze the hydraulics of RM2338. For instance, using Figure 5.13, one can extrapolate the maximum ponding depth under 100mm/hr (4in/hr) for 4-lane highway. Since few drivers attempt to drive fast under the heavy rainfall of 10in/hr, this value can be used for computing the plausibility of hydroplaning. In conclusion, the model application on RM 2338 Texas rural highway shows the typical pattern of sheet flow on superelevation transitions is consistent on real roads.

Chapter 7: Conclusions and Recommendations for Future Work

7.1 CONCLUSIONS

Gravity is the predominant driving force of sheet flow. Kinematic wave model is based on the assumption that gravity is the only driving force for flow. Due to the simplicity in theory and the reasonable accuracy of results, the kinematic wave model is popularly used in various hydrologic process modeling works including overland flow. However, this model is often inapplicable to urban hydraulics such as reverse flow, backwater effect, or converging (or diverging) flow in which non-gravity terms are as important as the gravity term. In sheet flow near the zero cross slope section on superelevation transitions, the pressure gradient becomes significant near the flat area. The diffusion wave model, with gravity and pressure gradient terms in the equation of motion, works well in modeling most urban hydraulic processes as well as other areas of interest.

This research focuses on building a numerical model with which one can simulate sheet flow on highway pavement surfaces of superelevation transitions under constant rainfall. The objectives of this research include the development of (1) an algorithm that creates a grid space of the physical domain defined by GEOPAK, (2) open boundary conditions using kinematic wave theory, (3) mathematical and numerical models for solving hydraulics of sheet flow, (4) an algorithm for drainage inlet placement, and (5) a FORTRAN code for implicit time-series computation of a system of equations.

A diffusion wave model is developed to simulate rainfall-runoff on nonporous surfaces. The model can compute unsteady (rising hydrograph) rainfall-runoff processes on the pavement surfaces of superelevation transitions. Based on the Saint-Venant equations, the proposed model solves unsteady sheet flow under uniform rainfall on nonporous surfaces. The partial differential equation in the mathematical model has no analytical solution due to a nonlinear combination of first and second order differential terms; therefore, a numerical model is developed to numerically approximate the solution using implicit finite volume method. Built with three point central differencing and Crank-Nicolson method, the numerical model has second order accuracy in space and time. An implicit iterative solver is constructed for matrix computations based on a general conjugate gradient method with Cholesky decomposition preconditioning technique which guarantees the coefficient matrix to be symmetrical and positive definite making the solver singularity-free during the matrix computation. The conjugate gradient solver converges quickly on 2-dimensional problems with large system of equations. The proposed model is verified by comparing a 1D flow result to the kinematic wave solution. Results on 2D flow show that the proposed diffusion wave model is capable of predicting the accumulation or spread of sheet flow near the location of zero cross slope on superelevation transitions.

Accurate representation of road surface is important in computing sheet flow over superelevation transitions. Any small changes in the road surface elevation between grid points may result in a dramatic change in the flow response. For instance, on a domain of S_x=0.01 and Δx=0.6m, the change in the surface elevation between neighboring grids is 6mm while the change in the water depth is 0.006mm at steady state. The algorithm proposed in this research defines a computational space in a curvilinear coordinates system. The grid generation processes are mathematically rigorous, conceptually consistent with highway design guidelines, and physically accurate. The model reads the surface profile from the geometry file exported from the original in GEOPAK. The geometry file specifies the three dimensional coordinates (x, y, z) of the road center and the ends of each lane at constantly intervaled stations. Using the points at the center of road and on the side edges, the model computes a series of curves that are continuously connected to each other. The location of computational grid points is then interpolated between neighboring stations over the entire width of the road. In the curvilinear coordinates system, the direction tangent to the center of road is defined as the longitudinal axis and the direction normal to the road direction is the transverse axis. The curvilinear grid space is later transformed into a rectangular computational space for numerical computations of sheet flow. The conveyance terms in the model enable the geometric characteristics of the physical space to be conserved in the computational space. Limitations exist in the model concerning applicable shape of road surface geometry as specified in Chapter 3.

The kinematic boundary conditions developed for open boundaries provide smooth continuous flow near the boundaries. The upstream boundary condition provides additional loading of inflow from upstream to the boundary cells as a source term (like rainfall). For highways with roadside curbs, a Neumann type closed boundary condition defines a no-through flow condition at the side boundaries. Additionally, zero-depth gradient boundary condition applies on the downstream open boundary in case gutter flow is developed along the curb. In case curb-opening inlets exist on the outside-lane edge near zero cross slope section, the boundary condition changes to critical flow condition at the location of the inlet.

The characteristics of sheet flow on superelevation transitions are investigated through a numerical experiment. Sensitivities of sheet flow on longitudinal slope, rainfall intensity, surface texture, road width, and the location of curb-opening inlet are studied based on the model results on numerous simulations. Finally, the model is applied on an existing Texas highway. The result of the numerical experiment and application are listed as follows:

1. An accumulation of sheet flow occurs near the zero cross slope section in the superelevation transition area due to a geometric change in cross slope. The maximum ponding depth is proportional to road width, rainfall intensity, and pavement texture (the rougher the surface, the deeper the ponding). The longitudinal slope and location of curb-opening inlet also affect the hydraulics on superelevation transition areas, but not as proportionally as the other variables.
2. On *Type-I* roads, the existence of stagnation point in the superelevation transition area affects the location of the maximum ponding depth especially on mild longitudinal slope surfaces. On steep roads, major accumulation of flow occurs on the inside lanes. On *Type-II* roads, the spread area of accumulated flow on the outside lanes gets larger as the longitudinal slope of the road becomes steeper; however, the maximum depth barely changes.
3. The optimal longitudinal slope is found in the range of 0.3% and 0.4% if the maximum ponding depth is of concern. Sections with smaller longitudinal slope have accumulated water on the outside-lane edge and those with larger longitudinal slopes have accumulation at the inside-lanes edge.
4. *Type-I* roads with steep longitudinal slopes have the maximum depth on the inside lanes. Vehicles driving downhill in the inside lanes encounter the ponded water. If a vehicle tries to reduce speed not before entering the curved section but after passing the zero cross slope section, the vehicle may experience hydroplaning. Taking an inside (near the center of road) lane and reducing the speed before entering a curved section may reduce the risk of accidents.
5. *Type-II* roads always have the maximum depth on the outside lanes. On these roads, vehicles traveling uphill on the outside lanes would meet the accumulated

water. A driver naturally brakes to reduce speed before entering a curvature. According to the model results, the vehicle may be traveling on heavily ponded area when the driver steps on the brake on these roads.

6. The existence of the heavily accumulated flow on the traffic lanes near the location of zero cross slope may cause differential drag forces on the traveling vehicles which can lead to loss of control.
7. A curb-opening inlet installed at the right location reduces the amount of sheet flow on the road surface. However, efficiency is limited because the accumulation on the road is mainly the result of the extended flow path which is due to the transition in cross slope.

7.2 RECOMMENDATIONS FOR FUTURE WORK

Flow responses on different design superelevation rates, design speeds, and radius of curvatures need to be studied to better understand the hydraulics, though they are not directly involved with the geometry of superelevation transitions. There are various types of drainage inlets popularly used in practice. A comprehensive research on the efficiency of different inlets on superelevation transition areas may benefit the highway engineers and customers, however, result shows that drainage inlets installed at the side of road have limited efficiency. A new drainage system which may be able to cut off the stream paths on the traffic lanes needs to be developed. It can be constructed along the road at the edge of lanes or at the center of the road in the area where superelevation transition influences the hydraulics. A lateral perforated pipe constructed beneath the road surface with porous pavement on it may enhance drainage. With additional change in coding, these suggested drainage related topics are all available for numerical simulation with the proposed model.

Appendix A: Algorithm for Iterative Method

A.1 MATRIX FACTORIZATION

The proposed numerical model adopts the Modified Incomplete Cholesky Decomposition and General Conjugate Gradient Method (MICCG) that Cooley (1992) used to find iterative solutions in his research. The preconditioning process generates a symmetrical positive definite preconditioned matrix which is called *M* matrix. This approximate preconditioning of *A* matrix in Equation 4.23 is defined as

$$\underline{M} = \underline{U}^T \underline{D}\underline{U} \tag{A1.1}$$

where $\underline{U}$ is an upper triangular matrix with non-zero values on the main diagonal and at off-diagonal locations where A matrix has non-zero values. The upper diagonal matrix of a simple 2D problem with 3*3 grid points is presented in Figure A1.1

$$\underline{U} = \begin{bmatrix} u_{11} & u_{12} & 0 & u_{14} & 0 & 0 & 0 & 0 & 0 \\ & u_{22} & u_{23} & 0 & u_{25} & 0 & 0 & 0 & 0 \\ & & u_{33} & 0 & 0 & u_{36} & 0 & 0 & 0 \\ & & & u_{44} & u_{45} & 0 & u_{47} & 0 & 0 \\ & & & & u_{55} & u_{56} & 0 & u_{58} & 0 \\ & & & & & u_{66} & 0 & 0 & u_{69} \\ & & & & & & u_{77} & u_{78} & 0 \\ & & & & & & & u_{88} & u_{89} \\ & & & & & & & & u_{99} \end{bmatrix} \tag{A1.2}$$

The $u_{i,j}$ on the off diagonals (i<j) are the conveyances along ξ and η directions on the domain which appear in $\underline{A}$ matrix at the same locations. $\underline{D}$ is a positive diagonal matrix with non-zero values on the main diagonal.

$$\underline{D} = \begin{bmatrix} 1/u_{11} & & & & & & & & \\ & 1/u_{22} & & & & & & & \\ & & 1/u_{33} & & & & & & \\ & & & 1/u_{44} & & & & & \\ & & & & 1/u_{55} & & & & \\ & & & & & 1/u_{66} & & & \\ & & & & & & 1/u_{77} & & \\ & & & & & & & 1/u_{88} & \\ & & & & & & & & 1/u_{99} \end{bmatrix} \qquad \text{(A1.3)}$$

For incomplete Cholesky factorization, the u_n values are defined such that the sum of the elements along a row of $\underline{M}$ matrix equals to the sum of the elements along the same row of $\underline{A}$.

$$\begin{aligned} u_{11} &= a_{11} \\ u_{22} &= a_{22} - \frac{u_{12}{}^{2}}{u_{11}} - \frac{u_{12}u_{14}}{u_{11}} \\ u_{33} &= a_{33} - \frac{u_{23}{}^{2}}{u_{22}} - \frac{u_{23}u_{25}}{u_{22}} \end{aligned} \qquad \text{(A1.4)}$$

The general algorithm can be expressed as (Cooley, 1992).

$$u_{i,j} = (1+\delta)a_{i,i} - \sum_{k=1}^{i-1} \frac{u_{ki}{}^{2}}{u_{kk}} - \sum_{j=1}^{i-1} f_{ji} - \sum_{j=i+1}^{N} f_{ij} \qquad \text{(A1.5)}$$

where f_{ij} is a fill-in entry of $\underline{M}$ defined by

$$f_{ij} = \begin{Bmatrix} \sum_{k=1}^{i-1} \frac{u_{ki}u_{kj}}{u_{kk}}, & where\ a_{ij} = 0 \\ 0, & where\ a_{ij} \neq 0 \end{Bmatrix} \qquad \text{(A1.6)}$$

Since $\underline{U}$ is an upper triangular matrix, $u_{ij}=0$ for $i>j$. The main diagonal entries (u_{ii}) are to be larger than zero. This is guaranteed by placing δ in Equation A1.5. Factorization starts with δ=0 and if $a_{ii} \leq 0$ are detected, the factorization stops and new δ value replaces old one to continue the decomposition. An empirical equation is used in Cooley (1992).

$$\delta_{new} = \frac{3}{2}\delta_{old} + 0.001 \tag{A1.7}$$

where the initial value of δold is zero. New value keeps increasing until $a_{ii}>0$ is detected, then the factorization resumes.

A.2 GENERAL CONJUGATE GRADIENT METHOD

The original matrix can be equated with the preconditioning matrix $\underline{M}$ by introducing a supplementary matrix $\underline{N}$.

$$\underline{A} = \underline{M} + \underline{N} \tag{A1.8}$$

Substituting this into Equation 4.22 gives

$$\underline{M}\,\underline{x} = \underline{f} - \underline{N}\,\underline{x} \tag{A1.9}$$

The iteration equation is defined such that

$$\underline{M}\,\underline{x}_{k+1} = \underline{f} - \underline{N}\,\underline{x}_k \tag{A1.10}$$

where k is iteration step. Now Equation A1.8 can be rewritten in an iteration equation form

$$\underline{N}_k = \underline{A}_k - \underline{M}_k \tag{A1.11}$$

Use this equation to rewrite Equation A1.10 with $\underline{N}$ removed.

$$\underline{M}\,\underline{x}_{k+1} = \underline{M}\,\underline{x}_k + \underline{f} - \underline{A}\,\underline{x}_k \tag{A1.12}$$

Note that $\underline{f} - \underline{A}\,\underline{x}_k = \underline{r}_k$ is the residual of the original system at kth iteration. Let $\underline{s}_k = \underline{x}_{k+1} - \underline{x}_k$ and rewrite the equation.

$$\underline{M}\,\underline{s}_k = \underline{r}_k \tag{A1.13}$$

Based on Equation A1.1, the solution of Equation A1.13 is obtained by

$$\begin{aligned} \underline{U}^T \underline{D}\,\underline{y}_k &= \underline{r}_k \\ \underline{U}\,\underline{s}_k &= \underline{y}_k \end{aligned} \tag{A1.14}$$

These equations can be solved using Gaussian elimination. In the first step, $\underline{y}_k = \left(\underline{U}^T \underline{D}\right)^{-1} \underline{r}_k$ is computed by forward elimination then a backward elimination solves $\underline{s}_k = \left(\underline{U}\right)^{-1} \underline{y}_k$. Conjugate gradient method is a 2nd order method because at each iteration the new change in $\underline{x}$ which is called $\underline{p}_k$ is calculated using the change from the prior iteration, in addition to the vector $\underline{s}_k$. The algorithm begins with initial assumption for the unknown vector.

$$\underline{x}_0 = 0 \text{, and } \underline{r}_0 = \underline{f} \tag{A1.15}$$

The following steps are executed starting with k=0:

$$\begin{aligned} \underline{s}_0 &= \underline{M}^{-1}\,\underline{r}_0 \\ \underline{p}_0 &= \underline{s}_0 \\ \alpha_k &= \frac{\underline{s}_0{}^T\,\underline{r}_0}{\underline{p}_0{}^T\,\underline{A}\underline{p}_0} \\ \underline{x}_1 &= \underline{x}_0 + \alpha_0\,\underline{p}_0 \\ \underline{r}_1 &= \underline{r}_0 - \alpha_0\,\underline{A}\underline{p}_0 \end{aligned} \tag{A1.16}$$

The iteration proceeds to the next step in the following algorithm.

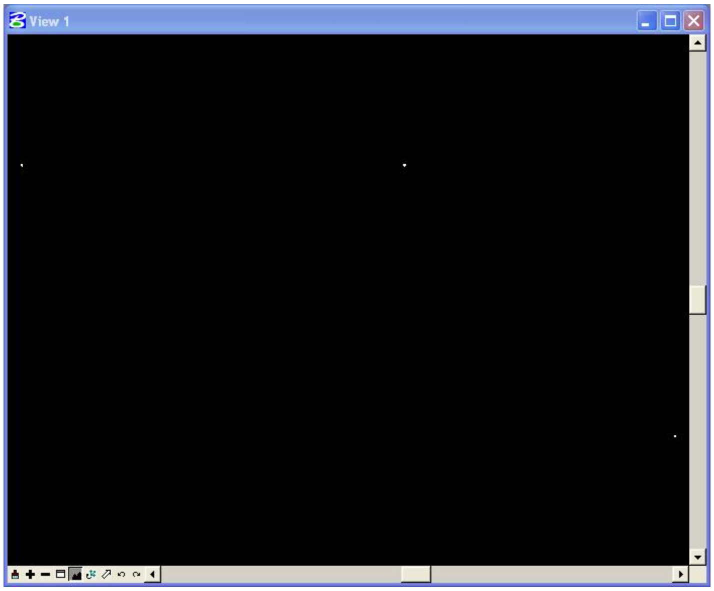
$$\left.\begin{aligned}
\underline{s}_k &= \underline{M}^{-1}\,\underline{r}_k \\
\beta_k &= \frac{\underline{s}_k^{\,T}\,\underline{r}_k}{\underline{s}_{k-1}^{\,T}\,\underline{r}_{k-1}} \\
\underline{p}_k &= \underline{s}_k + \beta_k\,\underline{p}_{k-1} \\
\alpha_k &= \frac{\underline{s}_k^{\,T}\,\underline{r}_k}{\underline{p}_k^{\,T}\,\underline{A}\,\underline{p}_k} \\
\underline{x}_{k+1} &= \underline{x}_k + \alpha_k\,\underline{p}_k \\
\underline{r}_{k+1} &= \underline{r}_k - \alpha_k\,\underline{A}\,\underline{p}_k
\end{aligned}\right\}\; k = 1,2,3,..... \tag{A1.17}$$

where the superscript *T* indicates the transpose of the vector.

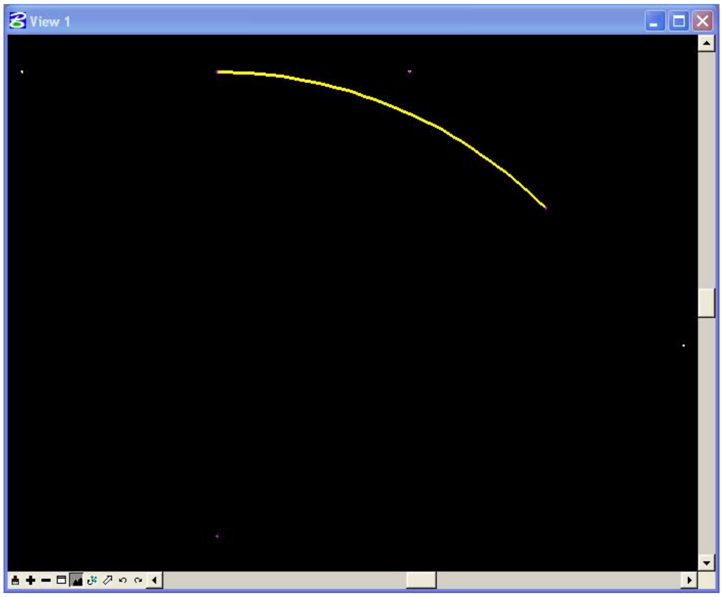
Appendix B: Creating Superelevation Transition in GEOPAK

The geometry files used in this research is designed in the GEOPAK and exported for the model application. In this chapter, an instruction to design a curved section of roadway using MicroStation GEOPAK is presented. The instruction provided in this chapter summarizes Dr. Thomas W. Rioux's class material of a transportation engineering course (http://www.ce.utexas.edu/prof/rioux/ce367_200501/GeometricDesignLab.htm).

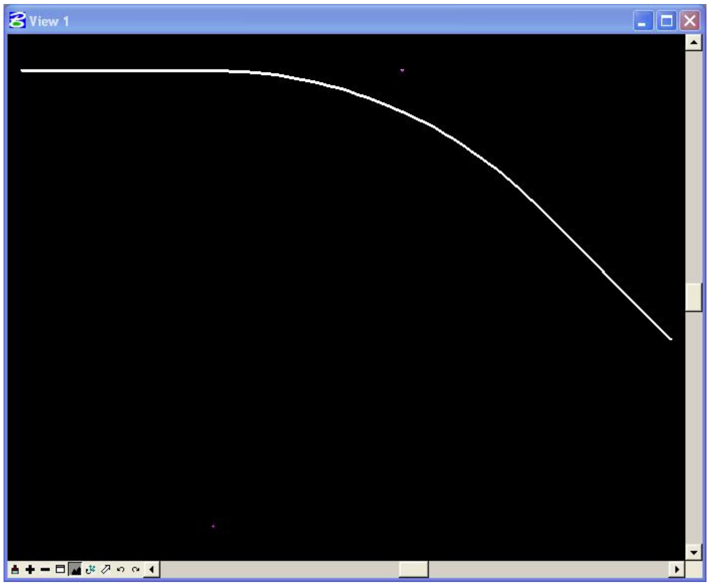
1. Start MicroStation. Make sure that the Working Units are feet with a resolution of 1000000 units per foot and that the Global Origin is zero (keyin "GO=$").
2. Start GEOPAK, set the standard GEOPAK user preferences for this class and set the standard COGO preferences for this class.
3. Create a GEOPAK Project named lab_07 using Job Number 07 with Subject of "Superelevation Runoff" and set Coordinate Geometry for Temporary Visualization.
4. Store the points for the centerline.
    - A. Store Point Number 1 at an X of 5000 and a Y of 5000.
    - B. Store Point Number 2 at a Bearing of N 90 E and a Distance of 1000 feet from Point Number 1.
    - C. Store Point Number 3 at a Bearing of S 45 E and a Distance of 1000 feet from Point Number 2.
    - D. Minimize the Coordinate Geometry dialog box. In MicroStation Window 1, choose the Fit View icon. The 3 points should be visible.



5. Store the horizontal circular curve by tangents using a Curve Name of CV1, Station set off, a Back Tangent with PB of 1 and PI of 2, Element set to Radius of 1200 feet, and an Ahead Tangent with Point Ahead (PA) of 3. Minimize the Coordinate Geometry dialog box. In MicroStation Window 1, choose the Fit View icon. The 3 points and the curve should be visible.



6. Create the centerline chain from elements with a Chain Name of CH1, Begin at 0+00, the 1st segment is Point 1, the 2nd segment is Curve CV1, and the 3rd segment is Point 3. Minimize the Coordinate Geometry dialog box. Minimize the Coordinate Geometry dialog box. In MicroStation Window 1, choose the Fit View icon. The 3 points, the curve, and the chain should be visible.



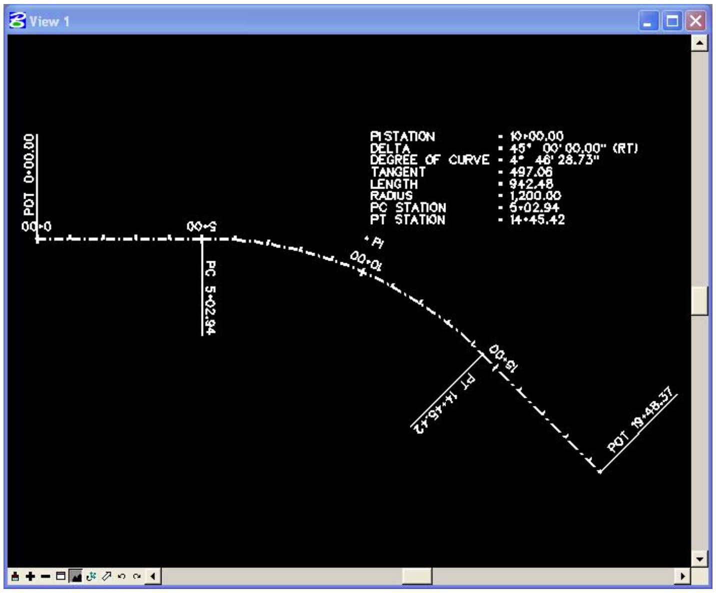
7. Draw the centerline chain and station the centerline chain.
   A. Draw the centerline chain using the Design and Computation Manager using the Texas Department of Transportation (TxDOT) GEOPAK parameter file "x:\microstation\txengd.ddb" selecting FEATURES then DRAFTING STANDARDS, then Alignments, then BL Baseline Horizontal Alignment, and finally Draw Plan & Profile for Job 07 setting Operation to Chains, Curve Data to on, all other options to off, and Label Scale to 200 for Chain CH1.
   B. Station the centerline chain setting Operation to Stationing, Tick Marks to on, Tick Marks Stations to on, PC & PT & CS... labels to on, PI labels to on, Small Ticks to Ticks LT; Labels LT, Large Ticks to Ticks Both; Labels LT, Control Point Labels to As Per Preferences, and Label Scale to 200 for Chain CH1.

C. Close the Plan & Profile Draw dialog box and close the Design and Computation Manager dialog box.

D. In the Coordinate Geometry dialog box, choose Tools -> Clear Visualized Elements (Temporary).

E. Move the Curve Data so the PI is visible.

F. In MicroStation Window 1, choose the Fit View icon. The centerline chain and stationing should be visible.

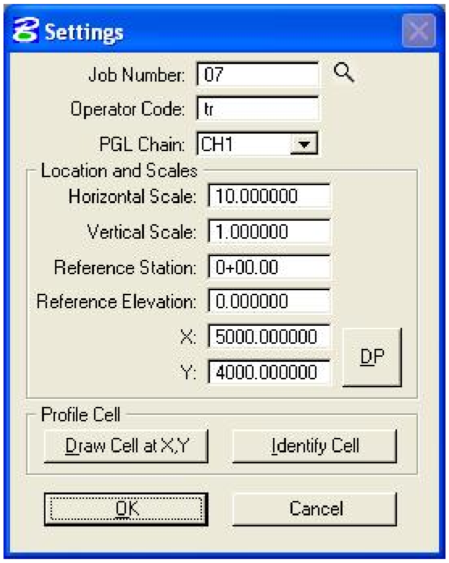


PI STATION = 10+00.00
DELTA = 45° 00' 00.00" (RT)
DEGREE OF CURVE = 4° 46' 28.73"
TANGENT = 497.06
LENGTH = 942.48
RADIUS = 1,200.00
PC STATION = 5+02.94
PT STATION = 14+45.42

8. Place a flat vertical profile at elevation 0.0.

A. Choose Applications -> GEOPAK ROAD -> Geometry -> Layout Profiles (VPI based).

B. In the Settings dialog box, set Job Number to 07, set Operator Code to your 2 initials, and set PGL Chain to CH1. In the Location and Scales group, set Horizontal Scale to 10.000000, set Vertical Scale to 1.000000, set Reference Station to 0+00.00, set Reference Elevation to 0.000000, set X to 5000.0, set Y to 4000.0, and press the OK button.



Job Number: 07

Operator Code: tr

PGL Chain: CH1

Horizontal Scale: 10.000000

Vertical Scale: 1.000000

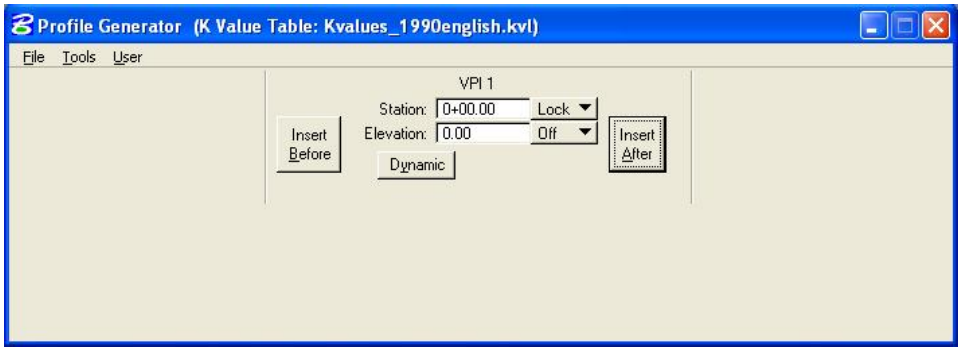
Reference Station: 0+00.00

Reference Elevation: 0.000000

X: 5000.000000

Y: 4000.000000

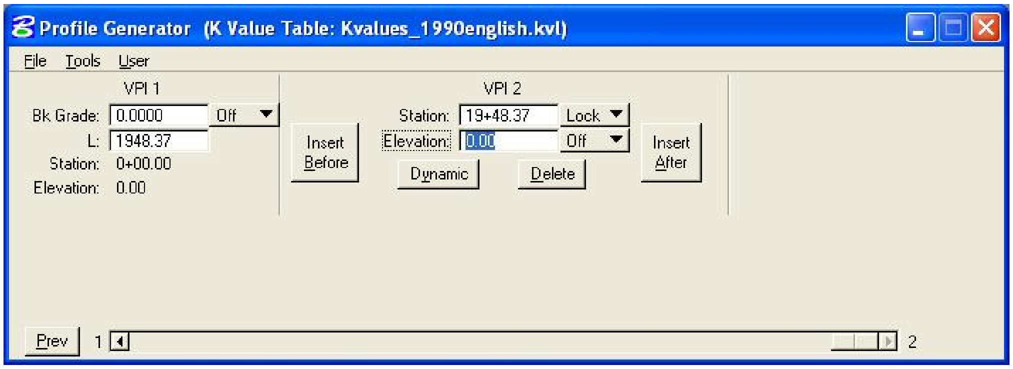
C. In the Profile Generator dialog box, set Station to 0+00.00 and set Elevation to 0.00.



Station: 0+00.00 Lock

Elevation: 0.00 Off

D. In the Profile Generator dialog box, press the Insert After push button. In the VPI 1 group, set Bk Grade to 0.0000 and press the Enter key and in the VPI 2 group, set Station to 19+48.37 (the POT at the end of the chain) and press the Enter key. In the VPI 1 group, the Length will be set to 1948.37 and in the VPI 2 group, the Elevation will be set to 0.00.


Profile Generator (K Value Table: Kvalues_1990english.kvl)

VPI 1

Bk Grade: 0.0000 Off

L: 1948.37

Station: 0+00.00

Elevation: 0.00

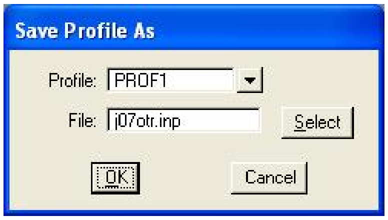
VPI 2

Station: 19+48.37 Lock

Elevation: 0.00 Off

E. In the Profile Generator dialog box, choose File -> Save As.

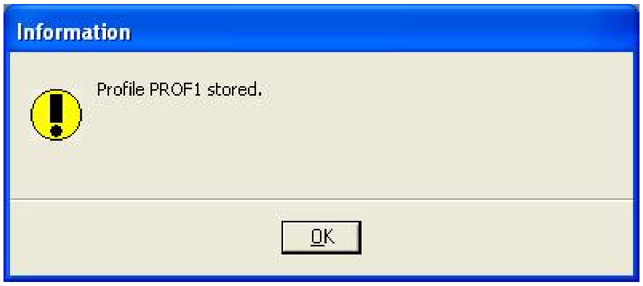
F. In the Save Profile As dialog box, set Profile to PROF1, set File to j07o<your_2_initials>.inp, and press the OK push button.


Save Profile As

Profile: PROF1

File: j07otr.inp

G. In the Information dialog box stating "Profile PROF1 stored", press the OK push button. Finally, close the Profile Generator dialog box. If an Alert dialog box stating "Do you want to save your profile?" appears, press the No push button.

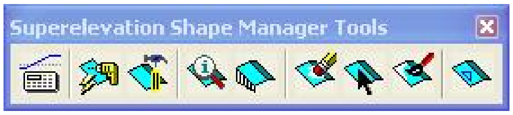

Information

Profile PROF1 stored.

9. Define the superelevation parameters.

   A. Choose Applications -> GEOPAK ROAD -> Cross Sections -> Superelevation Shape Manager Tools.

B. In the Superelevation Shape Manager Tools dialog box, choose the Automated Superelevation icon (leftmost icon) .

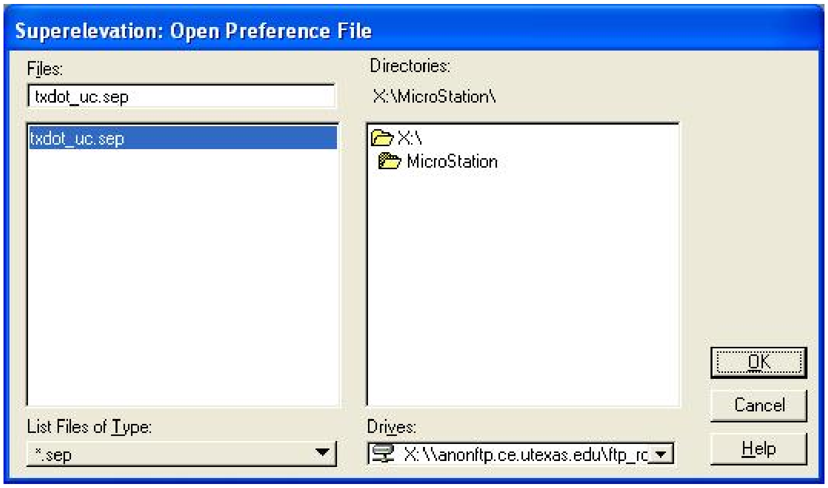


C. In the Automated Superelevation dialog box, choose File -> Directories, then press the Clear All push button, and finally press the OK push button.

D. In the Automated Superelevation dialog box, choose File -> Preferences.

E. In the Superelevation Preferences dialog box, choose File -> Open.

F. In the Superelevation: Open Preference File dialog box, under Drives, select X; under List Files of Type, select *.sep; under Directories, choose X:\MicroStation; under Files choose txdot_uc.sep, and press the OK push button.



Files: txdot_uc.sep

txdot_uc.sep

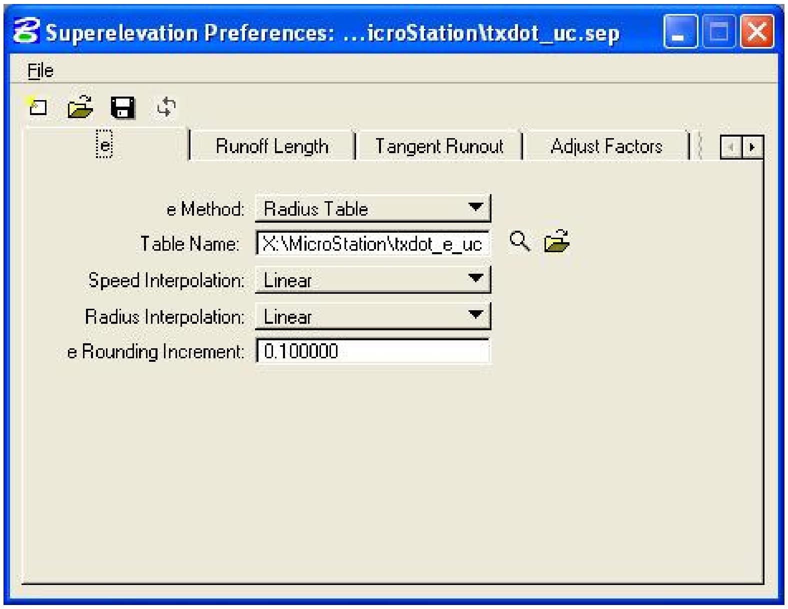
Directories: X:\MicroStation\

X:\
MicroStation

List Files of Type: *.sep

Drives: X: \\anonftp.ce.utexas.edu\ftp_rc

G. In the Superelevation Preferences dialog box, select the e tab and set e Method to Radius Table, the Table Name: should already be set to X:\MicroStation\txdot_e_uc.csv, set Speed Interpolation to Linear, set Radius Interpolation to Linear, and set e Rounding Increment to 0.100000.



e Method: Radius Table

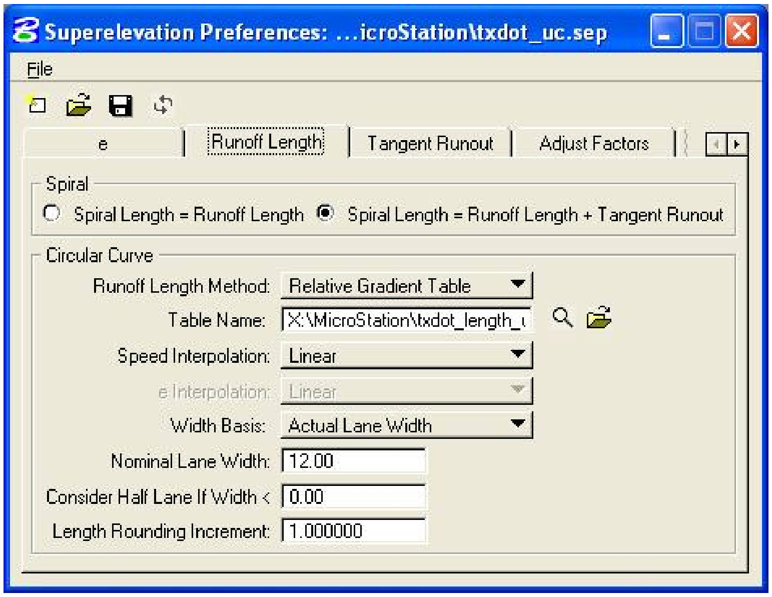
Table Name: X:\MicroStation\txdot_e_uc

Speed Interpolation: Linear

Radius Interpolation: Linear

e Rounding Increment: 0.100000

H. In the Superelevation Preferences dialog box, select the Runoff Length tab and select Spiral Length = Runoff Length + Tangent Runout, set Runoff Length Method to Relative Gradient Table, the Table Name should already be set to X:\MicroStation\txdot_length_uc.csv, set Speed Interpolation to Linear, set Width Basis to Actual Lane Width, set Nominal Lane Width to 12.00, set Consider Half Lane If Width < to 0.00, and set Length Rounding Increment to 1.000000.


Superelevation Preferences: ...icroStation\txdot_uc.sep

Spiral

○ Spiral Length = Runoff Length ● Spiral Length = Runoff Length + Tangent Runout

Circular Curve

Runoff Length Method: Relative Gradient Table

Table Name: X:\MicroStation\txdot_length_u

Speed Interpolation: Linear

e Interpolation: Linear

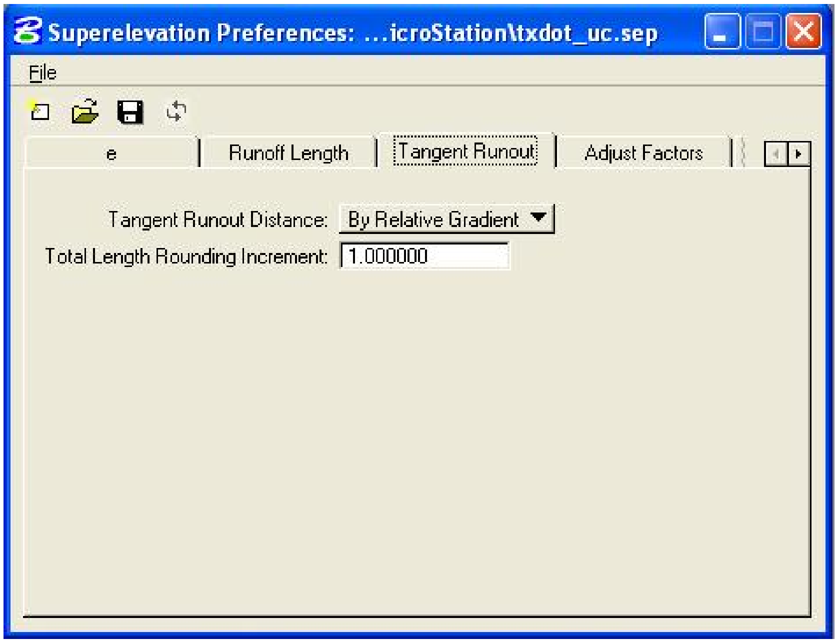
Width Basis: Actual Lane Width

Nominal Lane Width: 12.00

Consider Half Lane If Width < 0.00

Length Rounding Increment: 1.000000

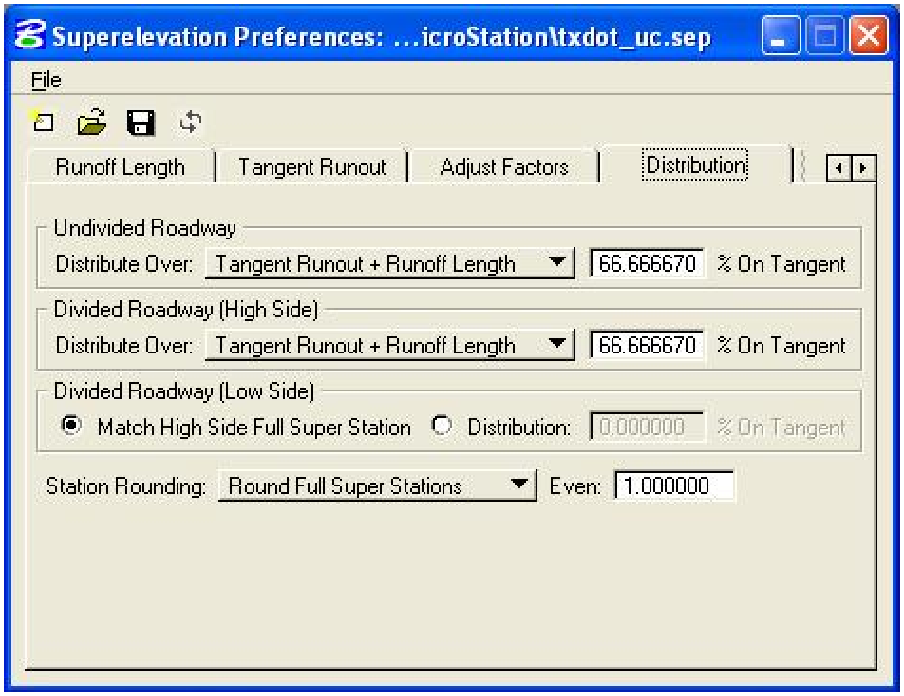
I. In the Superelevation Preferences dialog box, select the Tangent Runout tab and set Tangent Runout Distance to By Relative Gradient and set Total Length Rounding Increment to 1.000000.


Superelevation Preferences: ...icroStation\txdot_uc.sep

Tangent Runout Distance: By Relative Gradient

Total Length Rounding Increment: 1.000000

J. In the Superelevation Preferences dialog box, select the Distribution tab and set Undivided Roadway Distribute Over to Tangent Runout + Runoff Length with 66.666670 % On Tangent, set Divided Roadway (High Side) Distribute Over to Tangent Runout + Runoff Length with 66.666670 % On Tangent, set Divided Roadway (Low Side) Match High Side Full Super Station, and set Station Rounding to Round Full Super Stations to Even 1.000000.



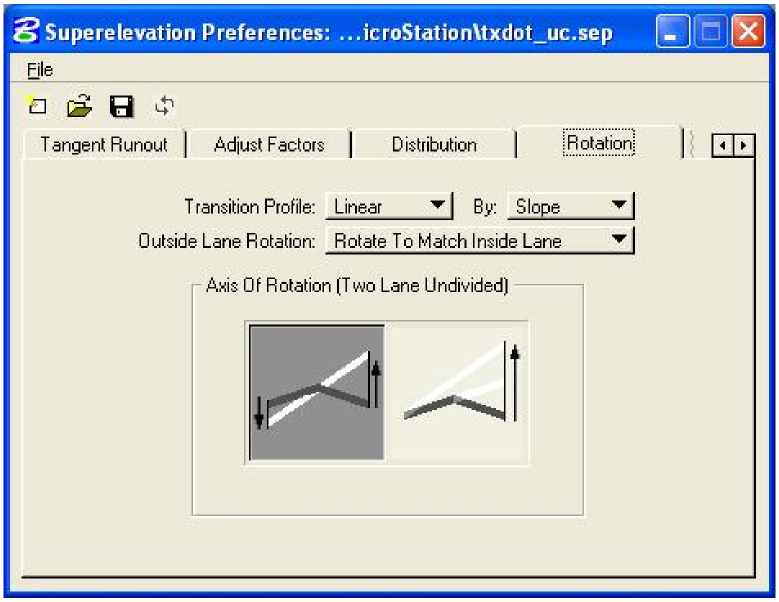
Undivided Roadway
Distribute Over: Tangent Runout + Runoff Length 66.666670 % On Tangent

Divided Roadway (High Side)
Distribute Over: Tangent Runout + Runoff Length 66.666670 % On Tangent

Divided Roadway (Low Side)
● Match High Side Full Super Station ○ Distribution: 0.000000 % On Tangent

Station Rounding: Round Full Super Stations Even: 1.000000

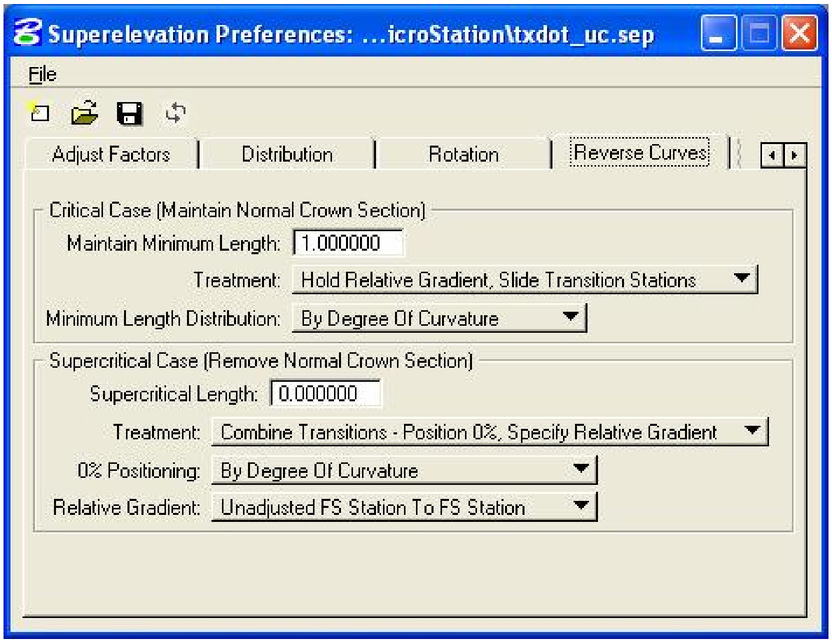
K. In the Superelevation Preferences dialog box, select the Rotation tab and set Transition Profile to Linear By Slope, set Outside Lane Rotation to Rotate To Match Inside Lane, and choose the left icon for Axis Of Rotation (Two Lane Undivided).


Superelevation Preferences: ...icroStation\txdot_uc.sep
File
Tangent Runout
Adjust Factors
Distribution
Rotation
Axis Of Rotation (Two Lane Undivided)

Transition Profile: Linear By: Slope

Outside Lane Rotation: Rotate To Match Inside Lane

L. In the Superelevation Preferences dialog box, select the Reverse Curves tab and in the Critical Case (Maintain Normal Crown Section) section set Maintain Minimum Length to 1.000000, set Treatment to Hold Relative Gradient, Slide Transition Stations, and set Minimum Length Distribution to By Degree Of Curvature; and in the Supercritical Case (Remove Normal Crown Section) section set Supercritical Length to 0.000000, set Treatment to Combine Transitions - Position 0%, Specify Relative Gradient, set 0% Positioning to By Degree Of Curvature, and set Relative Gradient to Unadjusted FS Station to FS Station.



Critical Case (Maintain Normal Crown Section)

Maintain Minimum Length: 1.000000

Treatment: Hold Relative Gradient, Slide Transition Stations

Minimum Length Distribution: By Degree Of Curvature

Supercritical Case (Remove Normal Crown Section)

Supercritical Length: 0.000000

Treatment: Combine Transitions - Position 0%, Specify Relative Gradient

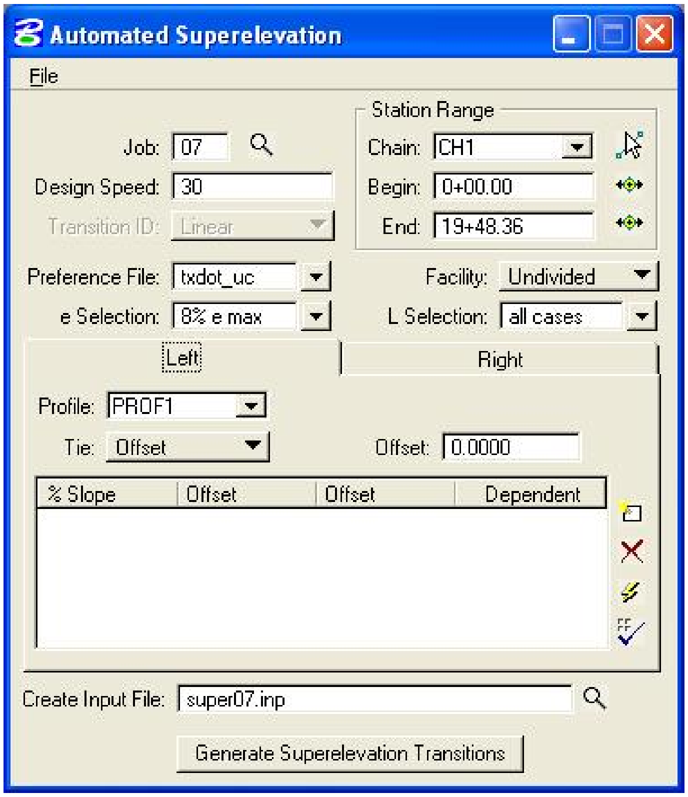
0% Positioning: By Degree Of Curvature

Relative Gradient: Unadjusted FS Station To FS Station

M. In the Superelevation Preferences dialog box, select File -> Save.

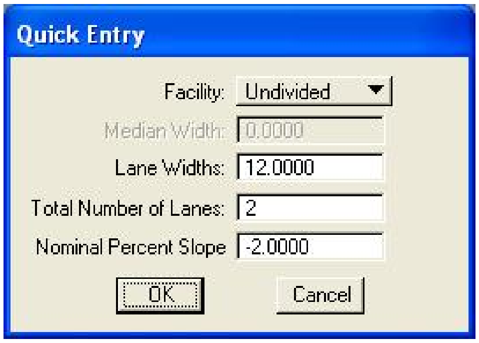
N. In the Superelevation Preferences dialog box, select File -> Exit.

O. In the Automated Superelevation dialog box, set Job to 07, set Design Speed to 30, set Preference File to txdot_uc, set e Selection to 8% e max, set Chain to CH1, set Begin to 0+00.00, set End to 19+48.36 (NOTE: this is 0.01 less than the POT value to avoid rounding issues in GEOPAK), set Facility to Undivided, and set L Selection to all cases. Press the Left tab and set Profile to PROF1, set Tie to Offset, set Offset to 0.0000, and set Create Input File to super07.inp.



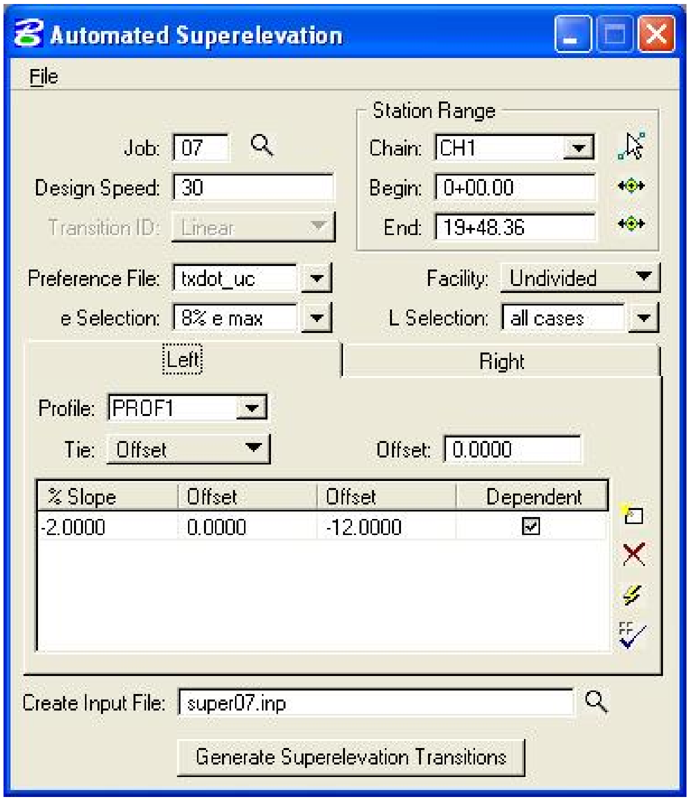
Job: 07
Design Speed: 30
Transition ID: Linear
Station Range
Chain: CH1
Begin: 0+00.00
End: 19+48.36
Preference File: txdot_uc
e Selection: 8% e max
Facility: Undivided
L Selection: all cases
Profile: PROF1
Tie: Offset
Offset: 0.0000
Create Input File: super07.inp

P. In the Automated Superelevation dialog box, press the Quick Entry icon ⚡. In the Quick Entry dialog box, set Facility to Undivided, set Lane Widths to 12.0000 feet, set Total Number of Lanes to 2, set Nominal Percent Slope to -2.0000, and finally press the OK push button.



Facility: Undivided
Median Width: 0.0000
Lane Widths: 12.0000
Total Number of Lanes: 2
Nominal Percent Slope: -2.0000

Q. In the Automated Superelevation dialog box, press the Left tab.



Job: 07

Design Speed: 30

Transition ID: Linear

Chain: CH1

Begin: 0+00.00

End: 19+48.36

Preference File: txdot_uc

Facility: Undivided

e Selection: 8% e max

L Selection: all cases

Profile: PROF1

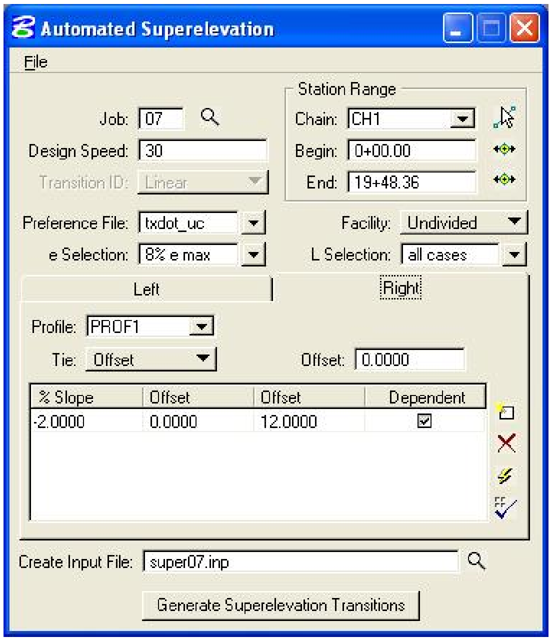
Tie: Offset

Offset: 0.0000

% Slope	Offset	Offset	Dependent
-2.0000	0.0000	-12.0000	☑

Create Input File: super07.inp

R. In the Automated Superelevation dialog box, press the Right tab.



Job: 07

Design Speed: 30

Transition ID: Linear

Chain: CH1

Begin: 0+00.00

End: 19+48.36

Preference File: txdot_uc

Facility: Undivided

e Selection: 8% e max

L Selection: all cases

Profile: PROF1

Tie: Offset

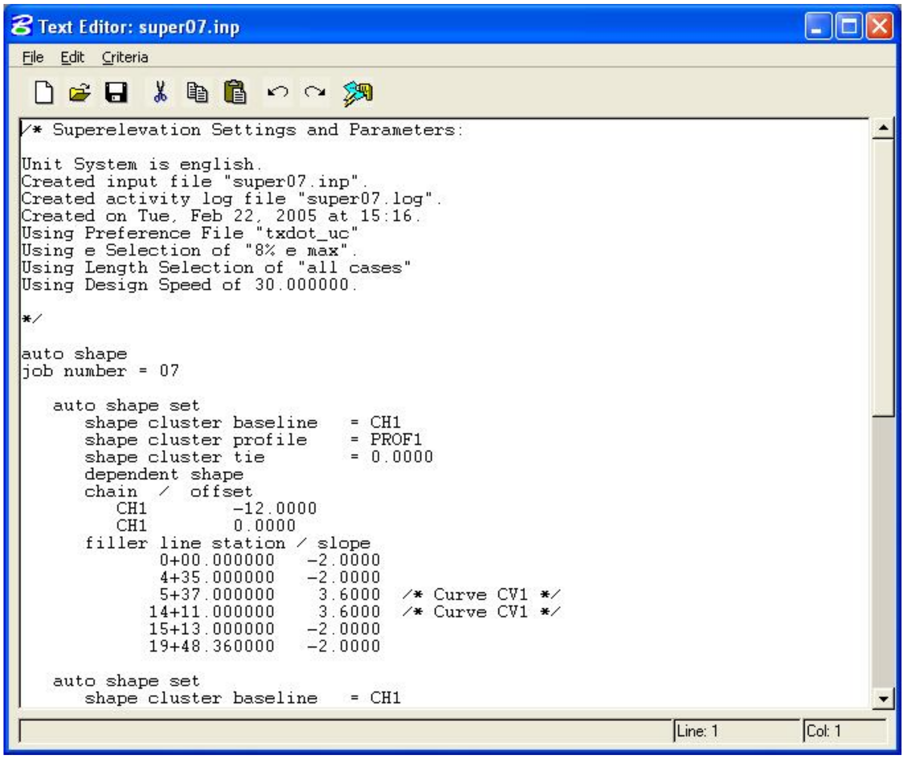
Offset: 0.0000

% Slope	Offset	Offset	Dependent
-2.0000	0.0000	12.0000	☑

Create Input File: super07.inp

S. In the Automated Superelevation dialog box, press the Generate Superelevation Transitions push button.

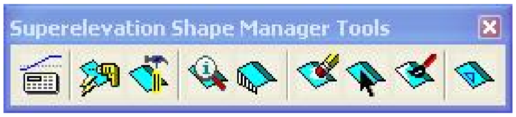
T. In the Text Editor: super07.inp dialog box, review the Superelevation Settings and Parameters (NOTE the filler line station values and slope values), close the Text Editor: super07.inp dialog box, and close the Automated Superelevation dialog box.



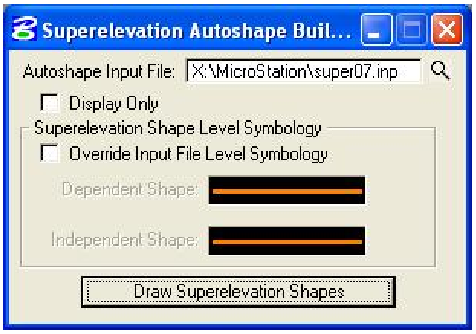
```
/* Superelevation Settings and Parameters:

Unit System is english.
Created input file "super07.inp".
Created activity log file "super07.log".
Created on Tue, Feb 22, 2005 at 15:16.
Using Preference File "txdot_uc"
Using e Selection of "8% e max".
Using Length Selection of "all cases"
Using Design Speed of 30.000000.

*/

auto shape
job number = 07

   auto shape set
      shape cluster baseline   = CH1
      shape cluster profile    = PROF1
      shape cluster tie        = 0.0000
      dependent shape
      chain  /  offset
         CH1         -12.0000
         CH1           0.0000
      filler line station / slope
             0+00.000000   -2.0000
             4+35.000000   -2.0000
             5+37.000000    3.6000   /* Curve CV1 */
            14+11.000000    3.6000   /* Curve CV1 */
            15+13.000000   -2.0000
            19+48.360000   -2.0000

   auto shape set
      shape cluster baseline   = CH1
```

10. Process the cross sections.

A. In the Superelevation Shape Manager Tools dialog box, choose the Autoshape Builder icon (2nd icon from the left).



B. In the Superelevation Autoshape Builder dialog box, set Autoshape Input File to X:\MicroStation\super07.inp, deselect Display Only, deselect Override Input File Level Symbology, press the Draw Superelevation Shapes push button, and close the Superelevation Autoshape Builder dialog box.



Superelevation Autoshape Buil...

Autoshape Input File: X:\MicroStation\super07.inp

☐ Display Only

Superelevation Shape Level Symbology

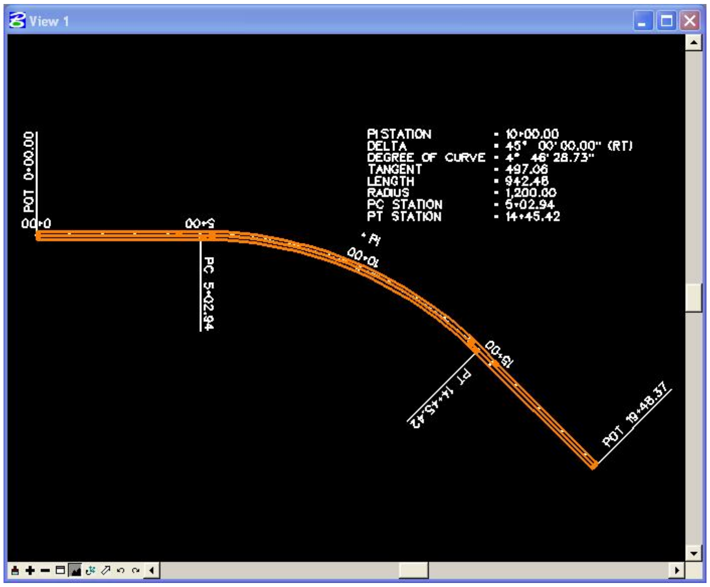
☐ Override Input File Level Symbology

Dependent Shape:

Independent Shape:

C. From the MicroStation dialog box, set the View Attributes for Fill to off for All views.

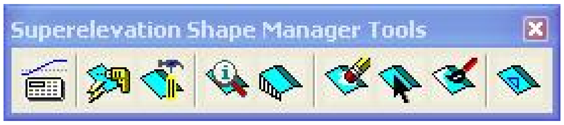
D. In MicroStation Window 1, choose the Fit View icon. The centerline chain and stationing and the superelevation shapes should be visible.



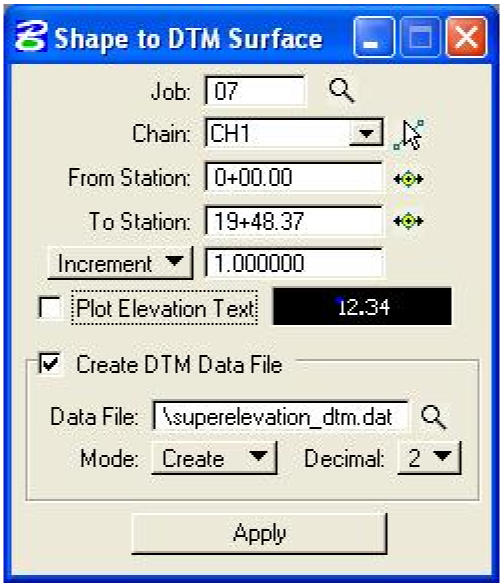
View 1

PI STATION = 10+00.00
DELTA = 45° 00' 00.00" (RT)
DEGREE OF CURVE = 4° 46' 28.73"
TANGENT = 497.06
LENGTH = 942.48
RADIUS = 1,200.00
PC STATION = 5+02.94
PT STATION = 14+45.42

11. In the Superelevation Shape Manager Tools dialog box, choose the Shape to DTM icon (rightmost icon)



A. In the Shape to DTM Surface dialog box, set Job to 07, set Chain to CH1, set From Station to 0+00.00 (should be automatically set when CH1 chosen), set To Station to 19+48.37 (should be automatically set when CH1 chosen), set Increment to 1.0, deselect Plot Elevation Text, select Create DTM Data File, press the magnifier glass icon to the right of the Data File entry box and choose a data file, set Mode to Create, set Decimal to 2, and finally press the Apply push button.


Shape to DTM Surface

Job: 07

Chain: CH1

From Station: 0+00.00

To Station: 19+48.37

Increment: 1.000000

☐ Plot Elevation Text 12.34

☑ Create DTM Data File

Data File: \superelevation_dtm.dat

Mode: Create

Decimal: 2

B. The Data File should now contain lines like the following:

```
1         5000.01         5012.00           -0.24     // 0+00.01 R 1
1         5000.01         5000.00            0.00     // 0+00.01 R 1
1         5000.01         4988.00           -0.24     // 0+00.01 R 1
1         5001.01         5012.00           -0.24     // 0+01.01 R 1
1         5001.01         5000.00            0.00     // 0+01.01 R 1
1         5001.01         4988.00           -0.24     // 0+01.01 R 1
```

Appendix C: FORTRAN Source Code and Output Files

C.1 SOURCE CODE

FORTRAN is one the most popular programming language in science and engineering. The latest version is FORTRAN 90/95 and it has adopted many good features of C++. 5 subprograms were coded separately and integrated into a computer model using the Lahey Developer v5.5. The list of the subprograms is as follows:

(1) main.f95
(2) cg.f95
(3) routines.f95
(4) files.f95
(5) post.f95

main.f95 is the main algorithm that integrates all the sub-routines and functions written in different programs. cg.f95 includes the MICCG solver for the penta-diagonal matrix system resulting from a 2D central differencing in space and a function which estimates the conveyance values across computational nodes. *routines.f95* represents the geometry algorithm. A curvilinear irregular grid space is generated in this subroutine based on the roadway profile specified in the DTM input file. *files.f95* and *post.f95* are the subroutines that are related to post-processing.

```
!       Last change:   JAEHAK JEONG     26 Sep 2007     10:52 am
!*********************************************************************************************
! DIFFUSION WAVE MODEL FOR STORMWATER RUNOFF ON HIGHWAY PAVEMENT SURFACES
!*********************************************************************************************

MODULE resource
        IMPLICIT REAL*8(a-h,o-z)
! Parameters
        REAL(KIND=8), PARAMETER :: asph_n=0.015 ! Asphalt pavement
        REAL(KIND=8), PARAMETER :: conc_n=0.012 ! Concrete pavement

        REAL(KIND=8), PARAMETER :: EPS0=0.01, EPS_SS=0.02, EPS_CG=1.0E-4, EPS_NL=1.0E-5
        REAL(KIND=8), PARAMETER :: PI=3.14159, g_e=32.2, g_m=9.81, phi_e=1.486
        REAL(KIND=8), PARAMETER :: nu_e=9.619e-6, nu_m=8.936e-7 ! [ft2/s],[m2/s] at 25C
        INTEGER, PARAMETER :: maxits=100, maxitsPic=20

        SAVE
! Geometric variables
          REAL(KIND=8), ALLOCATABLE, DIMENSION(:,:) :: xx, yy, zz ! GEOPAK data points
          REAL(KIND=8), ALLOCATABLE, DIMENSION(:,:) :: x, y, z       ! Grid point center
          REAL(KIND=8), ALLOCATABLE, DIMENSION(:,:) :: xb, yb, zb ! Grid block corners
          REAL(KIND=8), ALLOCATABLE, DIMENSION(:,:) :: Lx, Lb          ! Length of each cell
          REAL(KIND=8), ALLOCATABLE, DIMENSION(:,:) :: Sxh, sxv, Syh, Syv, Sfh, Sfv ! friction slope
          REAL(KIND=8), ALLOCATABLE, DIMENSION(:,:) :: Sox, Soy, So     ! surface slope/direction
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: alpha, thet, thet_DB    ! Roadway direction at stations
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: xcc, ycc, rc ! Center of circle
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: gamma             ! direction of roadway at each grid
point
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: xii                 ! longitudinal ratio of curve interpolation
0<xii<1
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: zc                  ! center line elevation
          CHARACTER(LEN=3), ALLOCATABLE, DIMENSION(:) :: shp          ! shape flag 'l' or 'c'
          CHARACTER(LEN=10), ALLOCATABLE, DIMENSION(:) :: station ! station of road section
        CHARACTER(LEN=20) :: filename, curve_pattern
        REAL(KIND=8) :: width, L_cum, mann, phi, sldr_width ! total width, Manning's coefficient
        INTEGER :: Ny_ctr, INx, INy, yi,ye, Nj, nin, izcs, j1grate, j2grate, j3grate

! Flow variables
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:,:) :: H_OLD, H_NEW, HT_OLD, HT_NEW, H_ITR
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:,:) :: qx, qy
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:,:) :: csx, csy, csx_old, csy_old, Diffx, Diffy
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: HT_DN, Stmp, rb, rb_old
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: hc_new, hc_old
        REAL(KIND=8) :: ri, h_min
! Misc
        REAL(KIND=8) :: dx, dy, dt, xi, eta, delta, resid1, resid2, norm2,xitv
          INTEGER ::   ii, jj, kk, ierror, ierror1, inum, Nx, Ny, iter_cg=0, NODES, t_ss
        INTEGER, DIMENSION(4,2) :: sb
        REAL(KIND=8) :: s1x, s1y, t_accum, t_ini
```

```
          REAL(KIND=8) :: norm2_cg=1, norm2_pic=1, norm2_ss=1, norm3_cg=1, norm3_pic=1, norm3_ss=1
          CHARACTER(LEN=10) :: date, time
          CHARACTER(LEN=2), ALLOCATABLE, DIMENSION(:) :: DNBC, LSBC, RSBC

! Flags
           INTEGER :: icurb   ! 1: roadside curbs exist, 0: no curb
          integer :: iunit   ! 1: SI unit                    0: English unit
          INTEGER :: inlet   ! 1: gutter inlet, 2: grate inlet, 0: no inlet

END module

!*****************************************************************************************

PROGRAM MAIN

          USE resource

          REAL :: time_start, time_finish, tc_tmp, qx_tmp, qy_tmp, angle
          INTEGER :: itr_pic=0, itr_ss=0

          INTERFACE
              SUBROUTINE output_files()
                  REAL :: dummy; END SUBROUTINE
              SUBROUTINE geometry()
                  REAL(KIND=8) :: ll, ddx, dxdxi, dydxi, dxc, dyc, drc, dthet
                  REAL(KIND=8) :: tmp1, tmp2, tmp3, xci, yci, ci, d, dx_adj, dy_adj
                  REAL(KIND=8) :: alpha1, alpha2, beta, L_sec, L_cum, thet1
                  INTEGER :: kkk, nn, ni, i, j ; END SUBROUTINE
              SUBROUTINE MICCG()
                  REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: aa, bb, cc, dd, ee, ff, uu, uu_old, p_old, p_new,
&
                  r_old, r_new, s_old, s_new, ym, tmp, alp, fi
                  REAL(KIND=8) :: rx, ry, alph, beta, sigma, del, resid3, resid4
                  INTEGER :: iflag, ll, nn ; END SUBROUTINE
              SUBROUTINE conveyance()
                  REAL(KIND=8) :: hmid, s1, s2, s3, s4, qh, qv ; END subroutine
              SUBROUTINE post(ti, tf)
                  REAL, INTENT(IN) :: ti, tf ; END SUBROUTINE
              SUBROUTINE upstream_BC()
                  REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: flow_path, Tc, qmax
                  CHARACTER(LEN=15) :: slope
                  REAL(KIND=8) :: sob, soxb, soyb, qb ; END SUBROUTINE
              SUBROUTINE down_BC()
                  REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: dl, dlx, dly, hs1
                  REAL(KIND=8) :: dli ; END SUBROUTINE
          END INTERFACE


! Read Input geometry file
!           WRITE(*,'(a24)') ' Input DTM file name :   '
!           READ(*,*) filename
          filename = 'rec01.dat'

!           WRITE(*,'(a24)') ' Terminal Time (sec) :   '
!           READ(*,*) t_ss
          t_ss = 300
          sldr_width = 0.
!           WRITE(*,'(a33)') ' English Unit(E) or SI Unit(S) :   '
!           READ(*,*) iunit

!           WRITE(*,'(a64)') ' no curb(0), curb inlet(1), grate inlet(2), curb no inlet(3): '
!           READ(*,*) icurb

!************************************************************
          iunit = 0 ! English unit

          icurb = 3 ! No curb
!           icurb = 1 ! curb inlet
!           icurb = 2 ! grate inlet
!           icurb = 3   ! curb with no inlet
!************************************************************

          mann =conc_n
          phi = phi_e


            OPEN (UNIT=3, FILE=filename, STATUS='old', IOSTAT=IERROR)

          ! Terminate the program if unable to read the file.
          if (ierror .NE. 0) then
                WRITE(*,*) " No source file found... Program terminated."
              call EXIT(0)          ! Terminate program
          END if
          ! Display source file on the screen
          WRITE(*,*)"
          WRITE(*,*)"
          WRITE(*,*)'*********************************************************************'
          WRITE(*,*)' Geometry Source : ', filename
          WRITE(*,*)'*********************************************************************'
          WRITE(*,*)"

! Define output files
          call output_files()

! Create computational grid space.
          call geometry()

          call cpu_time(time_start)
          ALLOCATE(H_OLD(Nx,Ny), H_NEW(Nx,Ny), HT_NEW(Nx,Ny), HT_OLD(Nx,Ny), H_ITR(Nx,Ny))
          ALLOCATE(qx(Nx,Ny), qy(Nx,Ny), DNBC(Ny),LSBC(Nx), RSBC(Nx))
```

```
        ALLOCATE(csx(Nx,Ny), csy(Nx,0:Ny), csx_old(Nx,Ny), csy_old(Nx,0:Ny))
        ALLOCATE(Diffx(Nx,Ny), Diffy(Nx,0:Ny), Stmp(Ny),   rb(Ny), rb_old(Ny))
        ALLOCATE(Sxh(Nx,Ny), Sxv(Nx,Ny), Sfh(Nx,Ny), Syh(Nx,Ny), Syv(Nx,Ny), Sfv(Nx,Ny))

! convert unit for rainfall intensity to [ft/sec]
        ri = 10.   ! [in/hr]
!        WRITE(*,*) 'Rainfall intensity [in/hr] : '
!        READ(*,*) ri
        ri = ri / 12. / 3600.
! Approximate the time of concentration
        tc_tmp = 1.0*(mann*L_cum/SQRT(MAXVAL(So)))**0.6 * (1./ri)**0.4
        WRITE(*,*)'Rough Estimate of Tc : ', tc_tmp/60., ' min'
        WRITE(*,*)''

! Number of cells within the domain
        IF(icurb.eq.0) then
           yi = 2; ye = Ny-1; Nj=Ny-2
        ELSE
           yi = 1; ye = Ny; Nj=Ny
        ENDIF
        NODES = (Nx-1) * Nj

! Default the variables
        csx=0; csy=0; csx_old=0; csy_old=0; kk=0; qx=0; qy=0; DNBC='AA'; LSBC='NF';RSBC='NF'
! Minimum water depth to initiate flow
        h_min = 0.000001
        qx = 0.00000001
        qy = 0.00000001
! Time for the minimum depth
        t_ini = h_min / ri
        t_accum = dt
!        t_accum = t_ini

! Initialize the time dependent variables.
        H_OLD = h_min; H_NEW = H_OLD; H_ITR = H_NEW; itr_ss=0
        HT_NEW = H_NEW + z; HT_OLD = HT_NEW; rb_old = 0

         OPEN (UNIT=30, FILE='temporalD.txt', STATUS='replace')

! Loop for time steps
        DO WHILE (t_accum<=t_ss+dt*0.01)
!        DO WHILE (norm2_ss>EPS_SS)

           call upstream_BC()
           call down_BC()
           if (icurb==0)   call side_BC()

       ! Iteration for computation of nonlinear diffusion coefficient
           norm2_pic=1; norm3_pic=1; itr_pic=0
```

```
!                 csx=.5; csy=.5
                DO WHILE(itr_pic<MAXITSPic)

                    call conveyance()

                    call MICCG()

                    norm2_pic = SQRT(SUM((H_NEW-H_ITR)**2)) / (SUM(H_NEW)/(Nx*Ny))
                    norm3_pic = MAXVAL(ABS(H_NEW-H_ITR)/H_NEW)

                    H_ITR = H_NEW
                    itr_pic = itr_pic + 1
                    WRITE(425,200) norm2_pic, norm3_pic
                    IF(norm2_pic<=EPS_NL.and.norm3_pic<=EPS_NL) EXIT

                END DO
          ! End Picard iteration

                norm2_ss = SQRT(SUM((H_NEW-H_OLD)**2)) / (SUM(H_NEW)/(Nx*Ny)*dt)
                norm3_ss = MAXVAL(ABS(H_NEW-H_OLD)/(dt*H_NEW))

                WRITE(425,*) ''
                WRITE(800,'(f10.5)') t_accum

! Update time dependent variabes
                H_OLD = H_NEW; HT_OLD = HT_NEW
                csx_old = csx; csy_old = csy
                rb_old = rb

! Print out results
       !         IF (MOD(itr_ss,10).eq.0) THEN
                    WRITE(*,300) itr_ss, ' #TIME[sec]:', t_accum, '#PIC:',itr_pic, '#CG:', iter_cg, 'h[mm]:',&
                             H_NEW(Nx/2,Ny/2)*304.8, norm2_ss, norm3_ss
      !          endif
                WRITE(450,200) norm2_ss, norm3_ss

                IF (MOD(itr_ss,1).eq.0) WRITE(200,100) (H_NEW(ii,Ny/2), ii=1,Nx)
                IF (MOD(itr_ss,50).eq.0) WRITE(30,150) t_accum, 304.8*H_NEW(Nx,Ny/2), 929.03*qx(Nx-1,Ny/2)
                itr_ss = itr_ss + 1
                t_accum = t_accum + dt

            END DO

            CALL cpu_time(time_finish)
            CALL post(time_start, time_finish)

            CLOSE(100); CLOSE(101); CLOSE(150); CLOSE(160); CLOSE(151); CLOSE(200)
            CLOSE(400); CLOSE(425); CLOSE(450); CLOSE(500); CLOSE(600); CLOSE(800)

100        FORMAT(30000f15.6)
```

```
150      FORMAT(f15.6, f15.5, f20.15)
200      FORMAT(2e20.5)
300      FORMAT(1x,I5,1X,a12,f8.3,3x,a5,I2,3x,a4,I3,3x,a6,f7.4,1x,2f10.4)

END PROGRAM

!***********************************************************************************************

 SUBROUTINE upstream_BC()

        USE resource
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: flow_path, Tc, qmax
        CHARACTER(LEN=15) :: slope
        REAL(KIND=8) :: sob, soxb, soyb, qb

        ALLOCATE(flow_path(Ny), Tc(Ny), qmax(Ny))

        IF(z(1,Ny_ctr)==z(1,1).and.z(1,Ny)==z(1,Ny_ctr)) then
           slope = 'flat'
        ELSEIF((z(1,1)-z(1,Ny_ctr))>0.and.(z(1,Ny_ctr)-z(1,Ny))>0) then
           slope = 'down-hill'
        ELSEIF((z(1,1)-z(1,Ny_ctr))<0.and.(z(1,Ny_ctr)-z(1,Ny))<0) then
           slope = 'up-hill'
        ELSE
           slope = 'normal-crown'
        ENDIF

        do jj=1,Ny
           eta = REAL(2.* jj - 1.) / (2.* Ny)
           soxb = Sox(1,jj)
           soyb = (zb(0,jj-1)-zb(0,jj)) / dy
           sob = SQRT(soxb**2 + soyb**2)
           IF(slope.eq.'normal-crown') then
              IF(eta==0.5) then
                 flow_path(jj) = 0.
              else
                 flow_path(jj) = ABS(eta-0.5) * (width+sldr_width*2) * ABS(sob / soyb)
              endif
           ELSEIF(slope.eq.'down-hill') then
              flow_path(jj) = eta * (width+sldr_width*2) * ABS(sob / soyb)
           ELSEIF(slope.eq.'up-hill') then
              flow_path(jj) = (1.-eta) * (width+sldr_width*2) * ABS(sob / soyb)
           else
              flow_path(jj) = 0
           endif
           qmax(jj) = ri * flow_path(jj)
           Tc(jj) = (mann / phi * flow_path(jj) / SQRT(sob)) ** 0.6 * (1./ri)**0.4
           qb = phi * SQRT(sob) / mann * (ri * t_accum) ** (5./3.)
           IF(t_accum<=Tc(jj)) then
              rb(jj) = qb * (soxb/sob)   / Lx(1,jj)
```

```
            ELSE
                rb(jj) = qmax(jj) * (soxb/sob) / Lx(1,jj)
            endif

        end do

  END SUBROUTINE

!*************************************************************************************

  SUBROUTINE down_BC()

!             dl : distance that a water particle moves from P1 to downstream boundary grid point during dt.
!             dlx : x component of dl
!             dly : y component of dl
!             hs1 : water depth at P1 at a time step (interpolated bilinearly)
!             cell_len : distance between P(Nx,j) and P(Nx-1,j)
!             dli : lateral distance from boundary to P(Nx,j)
!             slope : lateral shape of the downstream boundary

        USE resource
        REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: dl, dlx, dly, hs1, cell_len
        REAL(KIND=8) :: dli
        CHARACTER(LEN=15) :: slope


        ALLOCATE(dlx(Ny), dly(Ny), dl(Ny), hs1(Ny), cell_len(Ny))

        cell_len = (Lx(Nx,:) + Lx(Nx-1,:)) / 2.
        IF((z(Nx,Ny_ctr).EQ.z(Nx,1)).AND.(z(Nx,Ny_ctr).EQ.z(Nx,Ny))) then
            slope = 'flat'
        ELSEIF((z(Nx,1)-z(Nx,Ny_ctr))>0.and.(z(Nx,Ny_ctr)-z(Nx,Ny))>0) then
            slope = 'down-hill'
        ELSEIF((z(Nx,1)-z(Nx,Ny_ctr))<0.and.(z(Nx,Ny_ctr)-z(Nx,Ny))<0) then
            slope = 'up-hill'
        ELSE
            slope = 'crown'
        ENDIF

! Slope of free surface (total head gradient)
        DO jj=2,Ny-1
            IF(Soy(Nx,jj)<=0) then
                Stmp(jj) = (HT_OLD(Nx,jj)-HT_OLD(Nx,jj-1)) / dy
            ELSE
                Stmp(jj) = (HT_OLD(Nx,jj)-HT_OLD(Nx,jj+1)) / dy
            ENDIF
        END DO

! Zero Depth Gradient BC If water surface is flat,
! Kinematic BC if water surface is sloped.
```

```
     IF(slope.eq.'flat') then
        DNBC = 'KW'
     ELSE
        DNBC(1) = 'ZG'
        DNBC(Ny) = 'ZG'
        DO jj=2,Ny-1
           IF(Stmp(jj)<0.9*ABS(Soy(Nx,jj))) then ! 90% of bed slope is assumed gutter flow
              DNBC(jj) = 'ZG'          ! ZERO DEPTH GRADIENT BC
           ELSEIF(Sox(Nx,jj).eq.0) then
              DNBC(jj) = 'ZG'
           ELSE
              IF(DNBC(jj).eq.'ZG') THEN
                DNBC(jj) = 'ZG'
              ELSE
                DNBC(jj) = 'KW'          ! KINEMATIC BC
              ENDIF
           ENDIF
        END DO
     ENDIF

  ! Distance water flows on a characteristic curve during a time step
     dl = phi * SQRT(ABS(So(Nx,:))) / (mann * ri) * ((H_OLD(Nx,:)&
        + ri * dt)**(5./3.) - (H_OLD(Nx,:))**(5./3.))
     dlx = dl * ABS(COS(thet_DB))
     dly = dl * ABS(SIN(thet_DB))

     DO jj=1,Ny; IF(DNBC(jj).EQ.'KW') THEN

        s1x = 1. - MOD(dlx(jj),cell_len(jj)) / cell_len(jj)
        sb(1,1) = Nx - CEILING(dlx(jj)/cell_len(jj))
        sb(2,1) = Nx - FLOOR(dlx(jj)/cell_len(jj))
        sb(3,1) = sb(2,1)
        sb(4,1) = sb(1,1)

        IF(Soy(Nx,jj)>=0) then
           sb(1,2) = jj - CEILING(dly(jj)/dy)
           sb(2,2) = sb(1,2)
           sb(3,2) = jj - FLOOR(dly(jj)/dy)
           sb(4,2) = sb(3,2)
           s1y = 1 - MOD(dly(jj),dy) / dy
        else
           sb(1,2) = jj + FLOOR(dly(jj)/dy)
           sb(2,2) = sb(1,2)
           sb(3,2) = jj + CEILING(dly(jj)/dy)
           sb(4,2) = sb(3,2)
           s1y = MOD(dly(jj),dy) / dy
        endif
     ! Boundary points where stream lines come from outside of domain (trivial)
        IF(sb(1,2)>Ny.OR.sb(2,2)>Ny.OR.sb(3,2)>Ny.OR.sb(4,2)>Ny&
           .or.sb(1,2)<1.OR.sb(2,2)<1.OR.sb(3,2)<1.OR.sb(4,2)<1) THEN
```

```
              hs1(jj) = ri * dt
           ELSE
              hs1(jj) = (1.-s1x)*(1.-s1y)*(H_OLD(sb(1,1),sb(1,2))) &
                 + s1x*(1.-s1y)*(H_OLD(sb(2,1),sb(2,2))) &
                 + s1x*s1y * (H_OLD(sb(3,1),sb(3,2)))   &
                 + (1.-s1x)*s1y * (H_OLD(sb(4,1),sb(4,2)))
           ENDIF


        ! Check if P1 (upstream point) is out of boundary

          ! Lateral distance between grid point and laterally upslope boundary
           eta = REAL(2.* jj - 1.) / (2.* Ny)
           IF(slope.eq.'crown') then
              dli = (width+sldr_width*2) * ABS((eta-0.5))
           ELSEIF(slope.eq.'down-hill') then
              dli = (width+sldr_width*2) * eta
           ELSEIF(slope.eq.'up-hill') then
              dli = (width+sldr_width*2) * (1.-eta)
           else !flat
              dli = dly(jj)
           ENDIF

        ! Define ZDG BC if flow path goes out of boundary.
           IF(dly(jj)>dli.OR.(slope.eq.'crown'.and.eta.eq.0.5)) then
              DNBC(jj) = 'ZG'
           else
              H_NEW(Nx,jj) = (hs1(jj)**(5./3.) + (H_OLD(Nx,jj) + ri*dt)**(5./3.) &
                 - (H_OLD(Nx,jj))**(5./3.))**0.6
           ENDIF
        ENDIF; END DO
        DEALLOCATE(dl, dlx, dly)
 END SUBROUTINE
!*********************************************************************************************

subroutine side_BC()

        USE resource
        REAL(KIND=8) :: dli, dl, dlx, dly, hs1, grid_dist
        CHARACTER(LEN=15) :: slope



        IF(Soy(1,1) <0) RSBC(1) = 'ZG'
        IF(Soy(1,Ny)>0) LSBC(1) = 'ZG'

     ! Left side boundary
        do ii=2,Nx-1
           IF(Soy(ii,Ny)>0) LSBC(ii)='KW'
        end do
```

```
! The distance water flows on a characteristic curve in "dt"
    do ii=2,Nx-1; IF(LSBC(ii).eq.'KW') then
        dl = phi * SQRT(ABS(So(ii,Ny))) / (mann * ri) * ((H_OLD(ii,Ny)&
              + ri * dt)**(5./3.) - (H_OLD(ii,Ny))**(5./3.))
        dlx = dl * ABS(Sox(ii,Ny)/So(ii,Ny))
        dly = dl * ABS(Soy(ii,Ny)/So(ii,Ny))
        grid_dist = (Lx(ii,Ny) + Lx(ii-1,Ny)) / 2.

        s1x = 1. - MOD(dlx,grid_dist) / grid_dist
        sb(1,1) = ii - CEILING(dlx/grid_dist)
        sb(2,1) = ii - FLOOR(dlx/grid_dist)
        sb(3,1) = sb(1,1)
        sb(4,1) = sb(2,1)

        s1y = 1. - MOD(dly,dy) / dy
        sb(1,2) = Ny - CEILING(dly/dy)
        sb(2,2) = sb(1,2)
        sb(3,2) = Ny - FLOOR(dly/dy)
        sb(4,2) = sb(3,2)

! Boundary points where stream lines come from outside of domain (trivial)
        IF(sb(1,1)<1) THEN
            hs1 = ri * dt
        ELSE
            hs1 = (1.-s1x)*(1.-s1y)*(H_OLD(sb(1,1),sb(1,2))) &
                + s1x*(1.-s1y)*(H_OLD(sb(2,1),sb(2,2))) &
                + (1.-s1x)*s1y * (H_OLD(sb(3,1),sb(3,2)))    &
                + s1x*s1y * (H_OLD(sb(4,1),sb(4,2)))
        ENDIF

        H_NEW(ii,Ny) = (hs1**(5./3.) + (H_OLD(ii,Ny) + ri*dt)**(5./3.) &
               - (H_OLD(ii,Ny))**(5./3.))**0.6

    ENDIF; end do

! Right side boundary
    do ii=2,Nx-1
        IF(Soy(ii,1)<0) RSBC(ii)='KW'
    end do

! The distance water flows on a characteristic curve in "dt"
    do ii=2,Nx-1; IF(RSBC(ii).eq.'KW') then
        dl = phi * SQRT(ABS(So(ii,1))) / (mann * ri) * ((H_OLD(ii,1)&
              + ri * dt)**(5./3.) - (H_OLD(ii,1))**(5./3.))
        dlx = dl * ABS(Sox(ii,1)/So(ii,1))
        dly = dl * ABS(Soy(ii,1)/So(ii,1))
        grid_dist = (Lx(ii,1) + Lx(ii-1,1)) / 2.

        s1x = 1. - MOD(dlx,grid_dist) / grid_dist
        sb(1,1) = ii - CEILING(dlx/grid_dist)
```

```
          sb(2,1) = ii - FLOOR(dlx/grid_dist)
          sb(3,1) = sb(1,1)
          sb(4,1) = sb(2,1)

          s1y = 1. - MOD(dly,dy) / dy
          sb(1,2) = 1. + FLOOR(dly/dy)
          sb(2,2) = sb(1,2)
          sb(3,2) = 1. + CEILING(dly/dy)
          sb(4,2) = sb(3,2)

    ! Boundary points where stream lines come from outside of domain (trivial)
          IF(sb(1,1)<1) THEN
             hs1 = ri * dt
          ELSE
             hs1 = (1.-s1x)*(1.-s1y)*(H_OLD(sb(1,1),sb(1,2))) &
                + s1x*(1.-s1y)*(H_OLD(sb(2,1),sb(2,2))) &
                + (1.-s1x)*s1y * (H_OLD(sb(3,1),sb(3,2)))   &
                + s1x*s1y * (H_OLD(sb(4,1),sb(4,2)))
          ENDIF

          H_NEW(ii,1) = (hs1**(5./3.) + (H_OLD(ii,1) + ri*dt)**(5./3.) &
                - (H_OLD(ii,1))**(5./3.))**0.6

       ENDIF; end do
end subroutine

!       Last change:   JJ     26 Sep 2007     10:52 am
!**********************************************************************************************

subroutine geometry()
! This subroutine reads geometry data from a GEOPAK DTM file. Then generate a grid space
! for flow computation.
       USE resource

       REAL(KIND=8) :: ll, dxdxi, dydxi, dxc, dyc, drc, dthet
       REAL(KIND=8) :: tmp1, tmp2, tmp3, xci, yci, ci, d, dx_adj, dy_adj
       REAL(KIND=8) :: alpha1, alpha2, beta, L_sec,   thet1, S_inlet
       INTEGER :: kkk, nn, ni, i, j

       INTERFACE
          FUNCTION L_curv(a)
             INTEGER, INTENT(IN) :: a
             REAL(KIND=8) :: dxdy, dydx, L_curv ; END function
          FUNCTION thet_line(i,j)
             INTEGER, INTENT(IN) :: i, j
             REAL(KIND=8) :: thet_line ;   END FUNCTION
       END INTERFACE

   ! Check error and read the size of data points.
       READ(UNIT=3, FMT=1000, IOSTAT=ierror) tmp1, tmp2, tmp3
```

```
          if (ierror .eq. 0) then
               inum = 0
               DO WHILE(ierror1 .eq. 0)
                    inum = inum + 1
                    READ(UNIT=3, FMT=1000, IOSTAT=ierror1) tmp1, tmp2, tmp3
               END DO
          else
               WRITE(*,*) "Error in reading the data file"
             call EXIT(0)          ! Terminate program

          end if
          inum = INT(inum / 3.) - 1     ! 3 rows per each cross section. '0' to 'inum'

! Define dynamic variables
          ALLOCATE(xx(0:inum,3), yy(0:inum,3), zz(0:inum,3))
          ALLOCATE(xcc(0:inum), ycc(0:inum), rc(0:inum), thet(0:inum), alpha(1:inum-1))
          ALLOCATE(shp(0:inum), station(0:inum))

          xcc = 0; ycc = 0; rc = 0; thet = 0

! Read data points
          REWIND(UNIT=3)
          DO i=0,inum
               DO j=1, 3
                       READ(UNIT=3, FMT=1000, IOSTAT=ierror1) xx(i,j), yy(i,j), zz(i,j), station(i)
               END DO
          END DO
          CLOSE(UNIT=3)

          DO i=0,inum
             j = INDEX(station(i),'.')
             station(i) = ADJUSTR(station(i)(1:j-1))
          END DO

! Entire width of the roadway
          width = NINT(SQRT((xx(0,1)-xx(0,3))**2 + (yy(0,1)-yy(0,3))**2))
! Interval b/w stations (Lengthwise distance)
          xitv = NINT(SQRT((xx(0,1)-xx(1,1))**2 + (yy(0,1)-yy(1,1))**2))

! Alpha, the change in direction through neighboring points is
! evaluated along the center points.
! Alpha=0 for linear, Alpha=nonzero for curvilinear section.
! The average value of Alpha at y=1,2 and 3 will be taken to elimimate
! the noise in the data.
          alpha = 0
          DO i=1,inum-1
             do j=1,3
                if (ABS(xx(i,j)-xx(i-1,j)).gt.EPS0) then
                   alpha1 = ATAN((yy(i,j) - yy(i-1,j)) / (xx(i,j) - xx(i-1,j)))
```

```
                  if (xx(i,j).lt.xx(i-1,j)) alpha1 = alpha1 + pi
                  if (alpha1.lt.0) alpha1 = alpha1 + 2*pi
             else
                  if (yy(i,j).gt.yy(i-1,j)) then
                      alpha1 = pi / 2
                  else
                      alpha1 = pi * 3 / 2
                  end if
              end if

              if (ABS(xx(i+1,j)-xx(i,j)).gt.EPS0) then
                  alpha2 = ATAN((yy(i+1,j) - yy(i,j)) / (xx(i+1,j) - xx(i,j)))
                  if (xx(i+1,j).lt.xx(i,j)) alpha2 = alpha2 + pi
                  if (alpha2.lt.0) alpha2 = alpha2 + 2*pi
              else
                  if (yy(i+1,j).gt.yy(i,j)) then
                      alpha2 = pi / 2
                  else
                      alpha2 = pi * 3 / 2
                  end if
              end if

              if (alpha1<=pi/2 .and. alpha2>=1.5*pi) then
                  alpha(i) = alpha(i) + (alpha2 - 2*pi) - alpha1
              ELSEIF (alpha2<=pi/2 .and. alpha1>=1.5*pi) then
                  alpha(i) = alpha(i) + alpha2 - (alpha1 - 2*pi)
              else
                  alpha(i) = alpha(i) + alpha2 - alpha1
              end if
          end do
      END DO

! Eliminate noise in the data by taking the average value
! 1ft of lateral displacement in 1000ft length is neglected.
      alpha = alpha / 3.
      DO i=1,inum-1
          if (ABS(alpha(i)).lt.0.001) then
              alpha(i) = 0.
              shp(i) = 'L'
          else
              shp(i) = 'C'
          end if
      END DO

      shp(0) = shp(1)
      shp(inum) = shp(inum-1)

! define the transition point
      DO i=1,inum-1
          if (shp(i).eq.'C' .and. shp(i+1).eq.'L') then
```

```
                shp(i) = 'C-L'
            else if (shp(i-1).eq.'L' .and. shp(i).eq.'C') then
                shp(i) = 'L-C'
            end if
        END DO

    ! The pattern of curve (clockwise/counter-clockwise)
        IF(MINVAL(alpha)<0.and.MAXVAL(alpha)<=0) then
            curve_pattern = 'clockwise'
        ELSEIF(MINVAL(alpha)>=0.and.MAXVAL(alpha)>0) then
            curve_pattern = 'counter-clockwise'
        ELSEIF(MINVAL(alpha)==0.and.MAXVAL(alpha)==0) then
            curve_pattern = 'clockwise'!'straight line'
        ELSE
            WRITE(*,*) 'Error: Unable to define the curvatue shape!!'
            call exit(0)
        ENDIF

! Center of curve
        DO i=1,inum-1
            if (shp(i).ne.'L') then
                xci = (xx(i-1,2) + xx(i+1,2)) / 2
                yci = (yy(i-1,2) + yy(i+1,2)) / 2

                if (ABS(xx(i+1,2)-xx(i-1,2)).gt.EPS0) then
                    beta = ATAN((yy(i+1,2) - yy(i-1,2)) / (xx(i+1,2) - xx(i-1,2)))
                    if (xx(i-1,2)-xx(i+1,2).gt.EPS0) beta = beta + pi
                else
                    if (yy(i+1,2).gt.yy(i-1,2)) then
                        beta = pi / 2
                    else
                        beta = - pi / 2
                    endif
                endif

                ci = SQRT((xx(i+1,2) - xx(i-1,2)) ** 2 + (yy(i+1,2) - yy(i-1,2)) ** 2)
                rc(i) = ABS(ci / (2.* SIN(alpha(i))))

                IF(shp(i).eq.'L-C') rc(i) = rc(i+1)
                IF(shp(i).eq.'C-L') rc(i) = rc(i-1)
                d = ABS(rc(i) * COS(alpha(i)))

                if (curve_pattern.eq.'clockwise') then
                    xcc(i) = xci - d * COS(beta + pi/2)
                    ycc(i) = yci - d * SIN(beta + pi/2)
                else
                    xcc(i) = xci - d * COS(beta - pi/2)
                    ycc(i) = yci - d * SIN(beta - pi/2)
                end if
            end if
```

```
       END DO
       ! The first point and the last point
        if (shp(0).eq.'C') then
          xcc(0) = xcc(1)
         ycc(0) = ycc(1)
         rc(0) = rc(1)
        end if
        if (shp(inum).eq.'C') then
         xcc(inum) = xcc(inum-1)
         ycc(inum) = ycc(inum-1)
         rc(inum) = rc(inum-1)
        end if

! Roadway direction;
      ! Curved section, Direction of the radius of curvature &
      ! with respect to the horizontal axis
       DO i=0,inum
        IF(shp(i).ne.'L') then
            IF (ABS(xx(i,2)-xcc(i)).gt.EPS0) then
              thet(i) = ATAN((yy(i,2)-ycc(i))/(xx(i,2)-xcc(i)))
              IF(xx(i,2).lt.xcc(i)) thet(i) = thet(i) + pi
              IF(xx(i,2).gt.xcc(i).and.yy(i,2).lt.ycc(i)) thet(i) = thet(i) + 2*pi

            ELSEIF(ABS(xx(i,2)-xcc(i)).le.EPS0) then
              if (yy(i,2).gt.ycc(i)) then
                 thet(i) = pi / 2.
              else
                 thet(i) = -pi / 2.
              END if
            END IF
         END IF
       END DO

       DO i=1,inum-1
         IF(shp(i).eq.'L') then
      ! Linear section, Normal to the roadway direction
            thet(i) = thet_line(i-1,i+1)
         ENDIF
       END do
       do i=1,inum-1
         IF(shp(i).eq.'C-L') thet(i) = thet(i+1)
       end do
       IF(shp(1).eq.'L') thet(0) = thet(1)


       OPEN(1,FILE='geometric_data.txt')
       do i=0,inum
         WRITE(1,'(a5,6f20.5)') shp(i),xx(i,2),yy(i,2),rc(i),thet(i),xcc(i),ycc(i)
       end do
       CLOSE(1)
```

```
! Print out the curve information and get the grid scale
        WRITE(*,5000) 'Starting point : STA.', station(0)
        WRITE(*,5000) 'Ending     point : STA.', station(inum)
        WRITE(*,2000) 'Domain Length : ', xitv*inum, ' ft'
        WRITE(*,2000) 'Domain Width    : ', width, ' ft'
        WRITE(*,*) ''
        WRITE(*,3000) 'Curve Pattern : ', curve_pattern
        WRITE(*,*) ''
        WRITE(*,4000) 'Data Interval : ', xitv, ' ft'
        WRITE(*,*) ''
!        WRITE(*,*) ' Grid Intervals in X and Y :'
!        WRITE(*,*) ''
        WRITE(*,*) ' dx[ft], dy[ft], dt[sec] : '
        READ(*,*) dx, dy, dt
!        dx = 2.; dy = 2.; dt = 0.5
! Number of divisions for each section
        ni = NINT(xitv/dx)
        dx_adj = xitv / ni
        if (dx_adj .ne.dx) then
            WRITE(*,6000) 'dx (adjusted) :', dx_adj, 'ft'
            dx = dx_adj
            ni = NINT(xitv/dx)
        end if
          Nx = ni * inum

! Determine the latral size of refined mesh to an even number
        Ny = NINT((width+sldr_width*2)/dy)
      ! Force to put a grid point on the center line
        if (CEILING(Ny/2.).eq.Ny/2) Ny = Ny + 1
        dy_adj = (width+sldr_width*2) / Ny

! Grid number of the center point
        Ny_ctr = CEILING(Ny/2.)

        if (dy_adj .ne.dy) then
            WRITE(*,6000) 'dy (adjusted) :', dy_adj, 'ft'
            dy = dy_adj
        END if
        WRITE(*,*)''
! Define dynamic allocatable variables
        ALLOCATE(x(Nx,Ny),y(Nx,Ny),z(Nx,Ny),Lx(Nx,Ny),xii(0:ni),xb(0:Nx,0:Ny)&
                    ,yb(0:Nx,0:Ny),zb(0:Nx,0:Ny),Lb(Nx,0:Ny),Sox(Nx,Ny)&
                    ,Soy(Nx,Ny),So(Nx,Ny),thet_DB(Ny),gamma(Nx), zc(0:Nx))

! Grid point coordinates and cell length
    ! Upstream end
        DO j=0,Ny
            eta = REAL(j)/Ny
              IF (shp(0).eq.'L') then    ! line
```

```
                xb(0,j) = xx(0,2) - (eta-0.5) * (width+sldr_width*2) * SIN(thet(0))
                yb(0,j) = yy(0,2) + (eta-0.5) * (width+sldr_width*2) * COS(thet(0))
              ELSE ! curve
                if (curve_pattern.eq.'counter-clockwise') eta = 1. - eta
                xb(0,j) = xcc(0) + (rc(0) + (eta-0.5) * (width+sldr_width*2)) * COS(thet(0))
                yb(0,j) = ycc(0) + (rc(0) + (eta-0.5) * (width+sldr_width*2)) * SIN(thet(0))
              END IF
          END DO

      ! Within the domain
          DO i=0,inum-1 ! linear section
             IF ((shp(i).eq.'L').OR.(shp(i).eq.'C-L')) THEN
               call line_coord(i,ni)
             ELSE     ! curved section
               call curve_coord(i,ni)
             END IF
           END DO

   ! Smoothen Cell length
          DO ii=2,Nx-1
             Lb(ii,:) = (Lb(ii-1,:) + 4.*Lb(ii,:) + Lb(ii+1,:)) / 6.
          END DO
          DO jj=1,Ny
             Lx(:,jj) = (Lb(:,jj) + Lb(:,jj-1)) / 2.
          END DO

   ! Elevation of the grid center (cubic polynomial interpolation)
        ! Left side end
          call cubic_poly(1,Ny,ni)
        ! Right side end
          call cubic_poly(3,0,ni)
        ! Center line
          call cubic_poly(2,Ny_ctr,ni)

        ! upstream end
           zb(0,0) = zz(0,3); zb(0,Ny) = zz(0,1); zc(0) = zz(0,2)

        ! Extrapolate elevation at the shoulder edge
          zb(:,0) = zb(:,0) + (zb(:,0) - zc(:)) / (width / 2.) * sldr_width
          zb(:,Ny) = zb(:,Ny) + (zb(:,Ny) - zc(:)) / (width / 2.) * sldr_width

        ! Cell edges interpolation
          DO ii=0,Nx; DO jj=1,Ny-1
             eta = REAL(jj) / Ny
             IF(eta.le.0.5) then
                zb(ii,jj) = zb(ii,0) + (zc(ii) - zb(ii,0)) * (eta * 2.)
             ELSE
                zb(ii,jj) = zc(ii) + (zb(ii,Ny) - zc(ii)) * (eta * 2. - 1.)
             ENDIF
          END DO; END DO
        ! Cell center interpolation
         DO ii=1,Nx; DO jj=1,Ny
             z(ii,jj) = (zb(ii-1,jj-1) + zb(ii,jj-1) + zb(ii-1,jj) + zb(ii,jj)) / 4.
         END DO; END DO

! Surface slope
         do ii=1,Nx; do jj=1,Ny
             Sox(ii,jj)= -((zb(ii,jj)+zb(ii,jj-1))/2 &
                 - (zb(ii-1,jj)+zb(ii-1,jj-1))/2)/Lx(ii,jj)
             Soy(ii,jj)= -((zb(ii,jj)+zb(ii-1,jj))/2 &
                 - (zb(ii,jj-1)+zb(ii-1,jj-1))/2)/dy
         end DO; end do
         So = SQRT(Sox**2 + Soy**2)

! Direction of the steepest slope w.r.t. longitudinal direction
         thet_DB = ATAN(Soy(Nx,:) / Sox(Nx,:))

! Direction angle of roadway
         do i=0,inum-1; do kkk=1,ni
             ii = kkk + i * ni
             IF((shp(i).eq.'L').OR.(shp(i).eq.'C-L')) THEN
                 gamma(ii) = thet(i)
             ELSE
                 IF(curve_pattern.eq.'clockwise') THEN
                      gamma(ii) = thet(i) - pi / 2
                 ELSE
                     gamma(ii) = thet(i) + pi / 2
                 ENDIF
                 IF (gamma(ii).lt.0) gamma(ii) = gamma(ii) + 2*pi
                 IF (gamma(ii).gt.2*pi) gamma(ii) = gamma(ii) - 2*pi
             ENDIF
         end DO; END do

! Total length of the domain
         L_cum = xitv * inum

!*****************************************
! Locate zero-cross slope section
!*****************************************
         IF(icurb.ne.0) then
             IF(curve_pattern.eq.'clockwise') then
                 S_inlet = ABS(Soy(1,Ny))
                 do i=2,Nx
                     IF(ABS(Soy(i,Ny))<=S_inlet) then
                         S_inlet = ABS(Soy(i,Ny))
                         izcs = i
                     ENDIF
                  end do
             ELSEIF(curve_pattern.eq.'counter-clockwise') then
```

```
                S_inlet = ABS(Soy(1,1))
                do i=2,Nx
                    IF(ABS(Soy(i,1))<=S_inlet) THEN
                        S_inlet = ABS(Soy(i,1))
                        izcs = i
                    ENDIF
                end do
            ENDIF
        ENDIF

        IF(icurb.eq.1) then    ! Curb-opening inlet
         ! # of cells that face inlet opening
            nin = NINT(10./dx)
         ! distance from ZCS to down end of inlet
            izcs = izcs - (nin - 1) - 8
            ALLOCATE(hc_new(nin), hc_old(nin))
            hc_new = 0; hc_old = 0
        ELSEIF(icurb.eq.2) then      ! Grate inlet
         ! distance from ZCS to down end of inlet
            izcs = izcs - 8
            ALLOCATE(hc_new(3), hc_old(3))
            hc_new = 0; hc_old = 0
            IF(curve_pattern.eq.'clockwise') then
                j1grate = Ny - 2
                j2grate = Ny - 1
                j3grate = Ny
            ELSEIF(curve_pattern.eq.'counter-clockwise') then
                j1grate = 3
                j2grate = 2
                j3grate = 1
            ENDIF
        ENDIF

          DEALLOCATE(xx, yy, zz, zc)

1000      FORMAT (12X,3f20.4,5x,a10)
2000      FORMAT (1x,a17,f7.1,a3)
3000      FORMAT (1x,a17,a20)
4000      FORMAT (1x,a17,f5.1,a3)
5000      FORMAT (1x,a22,a10)
6000     FORMAT (1x,a16,f7.2,1x,a2)
100       FORMAT(1x,1000f15.5)
800      FORMAT(3f20.5)


! Surface elevation (Array)
        OPEN(111, FILE='elev.txt')
        do ii=1,Nx
            WRITE(111,800) x(ii,Ny_ctr), y(ii,Ny_ctr), z(ii,Ny_ctr)
        end DO
```

```
close (111)
end subroutine

!********************************************************************************************

FUNCTION thet_line(i,j)
! Calculate the direction angle of linear section of the roadway
        USE resource

        INTEGER, INTENT(IN) :: i, j
        REAL(KIND=8) :: thet_line

        IF (ABS(xx(j,2)-xx(i,2)).gt.EPS0) then
            thet_line = ATAN((yy(j,2)-yy(i,2)) / (xx(j,2)-xx(i,2)))
            if (xx(i,2).gt.xx(j,2)) thet_line = thet_line + pi
            if (thet_line.lt.0) thet_line = thet_line + 2 * pi
        ELSE
            if (yy(j,2).gt.yy(i,2)) then
                thet_line = pi / 2
            else
                thet_line = pi * 3 / 2
            end if
        END IF
END FUNCTION

!********************************************************************************************

subroutine curve_coord(i,ni)
! x and y coordinates of grid cells on the center (x,y) and edges (xb,yb) on curved section

        USE resource

        REAL(KIND=8) :: dxc, dyc, drc, dthet, rtmp, xct,yct, lseg, Lsum, thet_tmp
        INTEGER, INTENT (IN) :: i, ni
        INTEGER :: kkk, nn, j, k

        thet_tmp = thet(i+1)
        IF(shp(i+1).eq.'C-L') then
            IF(curve_pattern.eq.'clockwise') then
                thet_tmp = thet(i+1) + pi / 2.
            ELSEIF(curve_pattern.eq.'counter-clockwise') then
                thet_tmp = thet(i+1) - pi / 2.
            ENDIF
            if (thet_tmp.gt.2.*PI) thet_tmp = thet_tmp - 2.* PI
            if (thet_tmp.lt.0) thet_tmp = thet_tmp + 2.* PI
        ENDIF
```

```
          dxc = xcc(i+1) - xcc(i)
          dyc = ycc(i+1) - ycc(i)
          drc = rc(i+1) - rc(i)
          dthet = thet_tmp - thet(i)

          xii = 0.

! Determine segment increments.
          DO k=1,ni
             ii = i * ni + k
             lseg = 0.
             do WHILE (lseg<=dx)
                xii(k) = xii(k) + 1./ ni * 0.0001
                delta = thet(i) + xii(k)*dthet
                x(ii,Ny_ctr) = xcc(i) + xii(k)*dxc + (rc(i) + xii(k)*drc) * COS(delta)
                y(ii,Ny_ctr) = ycc(i) + xii(k)*dyc + (rc(i) + xii(k)*drc) * SIN(delta)
                IF(k==1) then
                   lseg = SQRT((x(ii,Ny_ctr)-xx(i,2))**2+(y(ii,Ny_ctr)-yy(i,2))**2)
                else
                   lseg = SQRT((x(ii,Ny_ctr)-x(ii-1,Ny_ctr))**2 &
                      +(y(ii,Ny_ctr)-y(ii-1,Ny_ctr))**2)
                endif
             END DO
             IF(k<ni) xii(k+1) = xii(k)
          END DO

! cell edges
          DO k=1,ni
             ii = i * ni + k
             xi = xii(k)
             delta = thet(i) + xi*dthet
             DO j=0,Ny
                eta = REAL(j)/Ny
                if (curve_pattern.eq.'counter-clockwise') eta = 1. - eta
                xb(ii,j) = xcc(i) + xi*dxc + (rc(i) + xi*drc &
                   + (eta-0.5)*(width+sldr_width*2)) * COS(delta)
                yb(ii,j) = ycc(i) + xi*dyc + (rc(i) + xi*drc &
                   + (eta-0.5)*(width+sldr_width*2)) * SIN(delta)
                Lb(ii,j) = SQRT((xb(ii,j)-xb(ii-1,j))**2+(yb(ii,j)-yb(ii-1,j))**2)
             END DO
          END DO

          do j=0,Ny
             Lsum = SUM(Lb(i*ni+1:(i+1)*ni,j))
             DO k=1,ni
                ii = i * ni + k
                Lb(ii,j) = Lsum / ni
             END do
          end do
```

```
 ! cell center point
         DO k=1,ni
            ii = i * ni + k
            DO j=1,Ny
               x(ii,j) = (xb(ii-1,j-1)+xb(ii-1,j)+xb(ii,j-1)+xb(ii,j)) / 4.
               y(ii,j) = (yb(ii-1,j-1)+yb(ii-1,j)+yb(ii,j-1)+yb(ii,j)) / 4.
               Lx(ii,j) = (Lb(ii,j-1)+Lb(ii,j))/2.
            END DO
         END DO

end subroutine

!****************************************************************************************

subroutine line_coord(i,ni)
! x and y coordinates of grid cells on the center (x,y) and edges (xb,yb) on linear section
         USE resource
         INTEGER, INTENT(IN) :: i, ni
         REAL(KIND=8) :: xlen

         xlen = SUM(SQRT((xx(i,:)-xx(i+1,:))**2 + (yy(i,:)-yy(i+1,:))**2)) / 3.

          DO kkk=1,ni
             ii = i * ni + kkk
            xi = REAL(kkk) / ni
            DO j=0,Ny
              eta = REAL(j) / Ny
              xb(ii,j) = xb(i*ni,j) + xi*xlen*COS(thet(i))
              yb(ii,j) = yb(i*ni,j) + xi*xlen*SIN(thet(i))
              Lb(ii,j) = SQRT((xb(ii,j)-xb(ii-1,j))**2+(yb(ii,j)-yb(ii-1,j))**2)
             END DO
            Lb(i*ni+1:(i+1)*ni,:) = xitv / ni
            Lx(i*ni+1:(i+1)*ni,:) = xitv / ni
          END DO

 ! cell center point
         DO k=1,ni
            ii = i * ni + k
            DO j=1,Ny
               x(ii,j) = (xb(ii-1,j-1)+xb(ii-1,j)+xb(ii,j-1)+xb(ii,j)) / 4.
               y(ii,j) = (yb(ii-1,j-1)+yb(ii-1,j)+yb(ii,j-1)+yb(ii,j)) / 4.
               Lx(ii,j) = (Lb(ii,j-1)+Lb(ii,j))/2.
            END DO
         END DO

end subroutine

!****************************************************************************************

subroutine cubic_poly(i1,i2,ni)
```

```
        USE resource
        INTEGER, INTENT(IN) :: i1, i2, ni
        REAL(KIND=8) :: slp1, slp2, c1, c2, c3, c4
        INTEGER :: k

        do i=0,inum-1
            IF(i==0) then
                slp1 = (zz(1,i1) - zz(0,i1))
                slp2 = (zz(2,i1) - zz(0,i1)) / 2.
            ELSEIF(i==inum-1) then
                slp1 = (zz(inum,i1) - zz(inum-2,i1)) / 2.
                slp2 = (zz(inum,i1) - zz(inum-1,i1))
            ELSE
                slp1 = (zz(i+1,i1) - zz(i-1,i1)) / 2.
                slp2 = (zz(i+2,i1) - zz(i,i1)) / 2.
            ENDIF

            do k=1,ni
                ii = k + i * ni
                xi = REAL(k) / ni
                c1 = -2.*(zz(i+1,i1) - zz(i,i1)) + slp1 + slp2
                c2 = 3.*(zz(i+1,i1) - zz(i,i1)) - (slp2 + 2.* slp1)
                c3 = slp1
                c4 = zz(i,i1)
                IF(i1.eq.2) then   ! center line
                    zc(ii) = c1 * xi ** 3 + c2 * xi ** 2 + c3 * xi + c4
                ELSE                   ! road sides
                    zb(ii,i2) = c1 * xi ** 3 + c2 * xi ** 2 + c3 * xi + c4
                ENDIF
            end do
        end do
end subroutine

!*******************************************************************************************



!       Last change:   JJ    18 Sep 2007      8:35 pm
!*******************************************************************************************
SUBROUTINE MICCG()
! This module solves Ax=f using a Modified Incomplete Cholesky Conjugate Gradient
! Method. The finite difference equation of interest is :
! a*H(i-1,j) + b*H(i,j-1) + c* H(i,j) + d*H(i,j+1) + e*H(i+1,j) = f(i,j)

    USE resource

    REAL(KIND=8), ALLOCATABLE, DIMENSION(:) :: aa, bb, cc, dd, ee, ff, uu, uu_old, &
    p_old, p_new, r_old, r_new, s_old, s_new, ym, tmp, alp, fi
    REAL(KIND=8) :: rx, ry, alph, beta, sigma, del, resid3, resid4, sfin
```

```
      INTEGER :: iflag, ll, nn, yend

      INTERFACE
         SUBROUTINE conveyance()
            REAL(KIND=8) :: hmid, s1, s2, s3, s4, qh, qv; END SUBROUTINE
      END INTERFACE

      ALLOCATE(aa(NODES),bb(NODES),cc(NODES),dd(NODES),ee(NODES),ff(NODES),uu(NODES) &
         ,uu_old(NODES),p_old(NODES),p_new(NODES),r_old(NODES),r_new(NODES) &
         ,s_old(NODES),s_new(NODES),ym(NODES),tmp(NODES),alp(NODES),fi(NODES))

      aa=0; bb=0; cc=0; dd=0; ee=0; ff=0; alp=0; iflag=0; del=0; fi=0; uu=0; iter_cg=1

!*********************************************
! Coefficient Matrix and RHS
!*********************************************
      DO ii=1,Nx-1
      IF(icurb==2.and.ii==izcs) THEN
         yend = ye-2
      ELSE
         yend = ye
      ENDIF
      DO jj=yi,ye
         IF(icurb==2.and.curve_pattern.eq.'clockwise') then
            IF(ii.LE.izcs) then
               nn = (ii-1) * Nj + (jj-yi+1)
            else
               nn = (ii-1) * Nj + (jj-yi+1-3)
            endif
         ELSEIF(icurb==2.and.curve_pattern.eq.'counter-clockwise') then
            IF(ii.LT.izcs) then
               nn = (ii-1) * Nj + (jj-yi+1)
            else
               nn = (ii-1) * Nj + (jj-yi+1-3)
            endif
         ELSE
            nn = (ii-1) * Nj + (jj-yi+1)
         ENDIF

!          IF(icurb==2) then
!          ELSEIF(icurb==2.and.curve_pattern.eq.'counter-clockwise') then
!          ELSE
!          ENDIF

         IF(ii/=1)      aa(nn) = - csx(ii-1,jj)
         IF(jj/=yi)     bb(nn) = - csy(ii,jj-1)
         IF(jj/=ye)     dd(nn) = - csy(ii,jj)
         IF(ii/=Nx-1) ee(nn) = - csx(ii,jj)

         cc(nn) = 2./ dt
```

```
        IF(ii/=1)       cc(nn) = cc(nn) + csx(ii-1,jj)
        IF(ii/=Nx-1) cc(nn) = cc(nn) + csx(ii,jj)
        IF(jj/=yi)     cc(nn) = cc(nn) + csy(ii,jj-1)
        IF(jj/=ye)     cc(nn) = cc(nn) + csy(ii,jj)

        ff(nn) = 2.* H_OLD(ii,jj) / dt + 2.* ri &
            - csx(ii,jj) * (z(ii,jj)-z(ii+1,jj)) &
            - csx_old(ii,jj) * (HT_OLD(ii,jj)-HT_OLD(ii+1,jj))

!          IF(ii==1) ff(nn) = ff(nn) + rb(jj) + rb_old(jj)

 ! No through flow on the upstream end
        IF(ii/=1) then
            ff(nn) = ff(nn) - csx(ii-1,jj) * (z(ii,jj)-z(ii-1,jj)) &
            - csx_old(ii-1,jj)*(HT_OLD(ii,jj)-HT_OLD(ii-1,jj))
        endif

 ! With roadside curbs
        IF(icurb.ne.0) then
            IF(jj/=yi) ff(nn) = ff(nn) - csy(ii,jj-1) * (z(ii,jj)-z(ii,jj-1)) &
                - csy_old(ii,jj-1) * (HT_OLD(ii,jj)-HT_OLD(ii,jj-1))
            IF(jj/=ye) ff(nn) = ff(nn) - csy(ii,jj) * (z(ii,jj)-z(ii,jj+1)) &
                - csy_old(ii,jj) * (HT_OLD(ii,jj)-HT_OLD(ii,jj+1))
        endif

! Critical depth B.C.(Dirichlet) for curb-opening inlet
        IF(icurb.eq.1) then
            hc_old = hc_new
            IF(ii>=izcs.and.ii<izcs+nin) then
                IF(curve_pattern.eq.'clockwise'.and.jj==ye) then
                    hc_new(ii-izcs+1) = 2./ 3. * H_NEW(ii,Ny)
                    Diffy(ii,Ny) = phi_e * hc_new(ii-izcs+1)**(5./3.) &
                        / (mann * ABS(Sxh(ii,Ny)))
                    csy(ii,Ny) = 2.* Diffy(ii,Ny) / dy**2
                    cc(nn) = cc(nn) + csy(ii,Ny)
                    ff(nn) = ff(nn) - csy(ii,Ny) * (z(ii,Ny) - zb(ii,Ny)) - csy_old(ii,Ny) &
                        * (HT_OLD(ii,Ny) - (zb(ii,Ny)+hc_old(ii-izcs+1))) + csy(ii,Ny) * hc_new(ii-izcs+1)
                ELSEIF(curve_pattern.eq.'counter-clockwise'.and.jj==yi) then
                    hc_new(ii-izcs+1) = 2./ 3. * H_NEW(ii,1)
                    sfin = (HT_NEW(ii,1) - (zb(ii,0) + hc_new(ii-izcs+1))) / (dy / 2.)
                    Diffy(ii,0) = phi_e * hc_new(ii-izcs+1)**(5./3.) &
                        / (mann * ABS(Sxh(ii,1)))
                    csy(ii,0) = 2.* Diffy(ii,0) / dy**2
                    cc(nn) = cc(nn) + csy(ii,0)
                    ff(nn) = ff(nn) - csy(ii,0) * (z(ii,1) - zb(ii,0)) - csy_old(ii,0) &
                        * (HT_OLD(ii,1) - (zb(ii,0)+hc_old(ii-izcs+1))) + csy(ii,0) * hc_new(ii-izcs+1)
                ENDIF
            ENDIF
        ENDIF
```

```
! Critical depth B.C. for grate inlet around sink cells (3ft(W) x 2ft(L))
        IF((icurb.eq.2).and.(ii.eq.izcs)) then
            hc_old = hc_new
            IF(curve_pattern.eq.'clockwise') then
                IF(ii.eq.izcs-1.and.jj.ge.ye-2) then
                    hc_new(3-(ye-jj)) = 2./ 3. * H_NEW(ii,jj)
                    Diffx(ii,jj) = phi_e * hc_new(3-(ye-jj))**(5./3.) &
                        / (mann * ABS(Sxh(ii,jj)))
                    csx(ii,jj) = 2.* Diffx(ii,jj) / Lx(ii,jj)**2
                    cc(nn) = cc(nn) + csx(ii,Ny)
                    ff(nn) = ff(nn) - csy(ii,Ny) * (z(ii,Ny) - zb(ii,Ny)) - csy_old(ii,Ny) &
                        * (HT_OLD(ii,Ny) - (zb(ii,Ny)+hc_old(ii-izcs+1))) + csy(ii,Ny) * hc_new(ii-izcs+1)
                ELSEIF(ii.eq.izcs.and.jj.eq.ye-3) then
                ELSEIF(ii.eq.izcs+1.and.jj.ge.ye-2) then
                ENDIF
            ELSEIF(curve_pattern.eq.'counter-clockwise'.and.jj==yi) then
                    hc_new(ii-izcs+1) = 2./ 3. * H_NEW(ii,1)
                    sfin = (HT_NEW(ii,1) - (zb(ii,0) + hc_new(ii-izcs+1))) / (dy / 2.)
                    Diffy(ii,0) = phi_e * hc_new(ii-izcs+1)**(5./3.) &
                        / (mann * ABS(Sxh(ii,1)))
                    csy(ii,0) = 2.* Diffy(ii,0) / dy**2
                    cc(nn) = cc(nn) + csy(ii,0)
                    ff(nn) = ff(nn) - csy(ii,0) * (z(ii,1) - zb(ii,0)) - csy_old(ii,0) &
                        * (HT_OLD(ii,1) - (zb(ii,0)+hc_old(ii-izcs+1))) + csy(ii,0) * hc_new(ii-izcs+1)
                ENDIF
        ENDIF

 ! no curb on sides
        IF(icurb.eq.0) then
            IF(ii==1.and.jj==yi.and.RSBC(1).eq.'ZG') ff(nn) = ff(nn) &
                - csy(ii,jj-1) * (z(ii,jj)-z(ii,jj-1)) &
                - csy_old(ii,jj-1) * (HT_OLD(ii,jj)-HT_OLD(ii,jj-1))
            IF(ii==1.and.jj==ye.and.LSBC(1).EQ.'ZG') ff(nn) = ff(nn) &
                - csy(ii,jj) * (z(ii,jj)-z(ii,jj+1)) &
                - csy_old(ii,jj) * (HT_OLD(ii,jj)-HT_OLD(ii,jj+1))
            IF(jj/=yi) ff(nn) = ff(nn) &
                - csy(ii,jj-1) * (z(ii,jj)-z(ii,jj-1)) &
                - csy_old(ii,jj-1) * (HT_OLD(ii,jj)-HT_OLD(ii,jj-1))
            IF(jj/=ye) ff(nn) = ff(nn) &
                - csy(ii,jj) * (z(ii,jj)-z(ii,jj+1)) &
                - csy_old(ii,jj) * (HT_OLD(ii,jj)-HT_OLD(ii,jj+1))
            IF(jj==yi.and.RSBC(ii).eq.'KW') then
                cc(nn) = cc(nn) + csy(ii,jj-1)
                ff(nn) = ff(nn) + csy(ii,jj-1) * H_NEW(ii,1) &
                    - csy(ii,jj-1) * (z(ii,jj)-z(ii,jj-1)) &
                    - csy_old(ii,jj-1) * (HT_OLD(ii,jj)-HT_OLD(ii,jj-1))
            ENDIF
            IF(jj==ye.and.LSBC(ii).eq.'KW') then
                cc(nn) = cc(nn) + csy(ii,jj)
                ff(nn) = ff(nn) + csy(ii,jj) * H_NEW(ii,Ny) &
```

```
                    - csy(ii,jj) * (z(ii,jj)-z(ii,jj+1)) &
                    - csy_old(ii,jj) * (HT_OLD(ii,jj)-HT_OLD(ii,jj+1))
                ENDIF
            endif

         ! Conditions for KW B.C. (downstream end)
            IF((DNBC(jj).EQ.'KW').AND.(ii.eq.Nx-1)) THEN
                cc(nn) = cc(nn) + csx(ii,jj)
                ff(nn) = ff(nn) + csx(ii,jj) * H_NEW(Nx,jj)
            ENDIF
        END DO; END DO

!*********************************************
! Incomplete Choleky Decomposition M=U'*D*U
!*********************************************

! U-Matrix main-diagonal
        do WHILE(iflag.eq.0)
            iflag = 1
            alp(1) = (1 + del) * cc(1)
            do ii=2,NODES
                alp(ii) = (1 + del)*cc(ii) - dd(ii-1)**2/alp(ii-1) &
                    - dd(ii-1)*ee(ii-1)/alp(ii-1)
                if (alp(ii)<=0.) then
                    del = 1.5 * del + 0.001
                    iflag = 0
                    exit
                end if
            end do
        end do

! Define the initial residual r_new = ff
        r_new = ff

!****************************************************
! General Conjugate Gradient method to solve (U'DU)x=ff
!****************************************************
        norm2_cg=1; norm3_cg=1
        do WHILE (iter_cg<=MAXITS)

        ! Forward substitution(UT*D*y=r where y=U*s) for y
            ym(1) = r_new(1)
            do ii=2,NODES
                ym(ii) = r_new(ii) - dd(ii-1) / alp(ii-1) * ym(ii-1)
                IF(ii>Nj) ym(ii) = ym(ii) - ee(ii-Nj) / alp(ii-Nj) * ym(ii-Nj)
            end do

        ! Backward substitution(DU*s=Dy)    for s
            ym = ym / alp
            s_new(NODES) = ym(NODES)
```

```
    DO ii=NODES-1,1,-1
        s_new(ii) = ym(ii) - dd(ii) / alp(ii) * s_new(ii+1)
        IF(ii<=Nodes-Nj) s_new(ii) = s_new(ii) &
            - ee(ii) / alp(ii) * s_new(ii+Nj)
    END DO

! Conjugate Gradient
    IF(iter_cg==1) then
        p_new = s_new
    else
        beta = SUM(s_new*r_new) / SUM(s_old*r_old)
        p_new = s_new + beta * p_old
    endif

    p_old=p_new; r_old=r_new; s_old=s_new; uu_old=uu

    do ii=1,NODES
        tmp(ii) = cc(ii) * p_new(ii)
        IF(ii>Nj)             tmp(ii) = tmp(ii) + aa(ii) * p_new(ii-Nj)
        IF(ii>1)              tmp(ii) = tmp(ii) + bb(ii) * p_new(ii-1)
        IF(ii<NODES)          tmp(ii) = tmp(ii) + dd(ii) * p_new(ii+1)
        IF(ii<=NODES-Nj) tmp(ii) = tmp(ii) + ee(ii) * p_new(ii+Nj)
    end do

    alph   = SUM(s_new*r_new) / SUM(p_new*tmp)
    uu       = uu + alph * p_new
    r_new = r_new - alph * tmp

    norm2_cg = SQRT(SUM((uu-uu_old)**2)) / (SUM(uu)/NODES)
    norm3_cg = MAXVAL(ABS(uu-uu_old)/uu)
    WRITE(400,1000) norm2_cg, norm3_cg

    if (norm2_cg<EPS_CG.AND.norm3_cg<=EPS_CG) then
        WRITE(400,*) ''
        exit
    else
        iter_cg = iter_cg + 1
    END if
END do

IF (iter_cg>MAXITS) THEN
    WRITE(*,*) 'ICOUNT exceeds MAXITS'
    call exit(0)
ELSE
    DO ii=1,Nx-1; DO jj=yi,ye
        nn = (ii-1) * Nj + (jj-yi+1)
        H_NEW(ii,jj) = uu(nn)
    END DO; END DO
END IF
```

```
     do jj=1,Ny
         IF(DNBC(jj).EQ.'ZG') H_NEW(Nx,jj) = H_NEW(Nx-1,jj)
     end do

     IF(LSBC(1).eq.'ZG') H_NEW(1,Ny) = H_NEW(1,Ny-1)
     IF(RSBC(1).eq.'ZG') H_NEW(1,1)    = H_NEW(1,2)

     do ii=1,Nx-1
         IF(LSBC(ii).eq.'NF') then
             H_NEW(ii,Ny) = H_NEW(ii,Ny-1) + z(ii,Ny-1) - z(ii,Ny)
         ENDIF
         IF(RSBC(ii).eq.'NF')THEN
             H_NEW(ii,1)    = H_NEW(ii,2) + z(ii,2) - z(ii,1)
         endif
     end do
 ! Restrict the depth not to be negative.
     do ii=1,Nx; do jj=1,Ny
         IF(H_NEW(ii,jj)<0.) H_NEW(ii,jj) = ri * dt
     end DO; end do

! Update total head
     HT_NEW = H_NEW + z

! Update Unit flow rate at the cell interfaces
     do ii=1,Nx-1
         qx(ii,1:Ny) = csx(ii,1:Ny) * (HT_NEW(ii,1:Ny) - HT_NEW(ii+1,1:Ny))
     end DO
     do jj=1,Ny-1
         qy(1:Nx-1,jj) = csy(1:Nx-1,jj) * (HT_NEW(1:Nx-1,jj) - HT_NEW(1:Nx-1,jj+1))
     end DO

     DEALLOCATE(aa,bb,cc,dd,ee,ff,uu,p_old,p_new,r_old,r_new,s_old&
                 ,s_new,ym,tmp,alp,fi)
1000      FORMAT(2e20.3)

END SUBROUTINE

!*************************************************************************************

SUBROUTINE conveyance()
! n=ne+c1/Re, Re=q/nu, q=phi/ne*h^5/3*S^1/2-c1*nu/ne, D=phi*h^5/3/(n*S^1/2)
          USE resource
          REAL(KIND=8) :: hmid, s1, s2, s3, s4, qh, qv, hc, zin

! Friction slope at the interface b/w longitudinally neighboring cells
         DO ii=1,Nx-1; DO jj=1,Ny
              Sxh(ii,jj) = (HT_NEW(ii,jj)-HT_NEW(ii+1,jj)) / ((Lx(ii,jj)+Lx(ii+1,jj)) / 2.)
              IF(jj/=1) then
                  s1 = (HT_NEW(ii,jj-1) - HT_NEW(ii,jj)) / dy
                  s2 = (HT_NEW(ii+1,jj-1) - HT_NEW(ii+1,jj)) / dy
```

```
            ENDIF

            IF(jj/=Ny) then
                s3 = (HT_NEW(ii,jj) - HT_NEW(ii,jj+1)) / dy
                s4 = (HT_NEW(ii+1,jj) - HT_NEW(ii+1,jj+1)) / dy
            ENDIF

            IF(jj==1) then
                Syh(ii,jj) = (s3 * Lx(ii+1,jj) + s4 * Lx(ii,jj)) / (Lx(ii,jj)+Lx(ii+1,jj))
            ELSEIF(jj==Ny) then
                Syh(ii,jj) = (s1 * Lx(ii+1,jj) + s2 * Lx(ii,jj)) / (Lx(ii,jj)+Lx(ii+1,jj))
            ELSE
                Syh(ii,jj) = ((s1+s3) * Lx(ii+1,jj) + (s2+s4) * Lx(ii,jj)) &
                    / (2.*(Lx(ii,jj)+Lx(ii+1,jj)))
            ENDIF
            Sfh(ii,jj) = SQRT(sxh(ii,jj)**2 + syh(ii,jj)**2)
        END DO; END DO

! Friction slope at the interface b/w laterally neighboring cells
        DO ii=1,Nx-1; DO jj=1,Ny-1
            Syv(ii,jj) = (HT_NEW(ii,jj)-HT_NEW(ii,jj+1)) / dy
            IF(ii/=1) then
                s1 = (HT_NEW(ii-1,jj) - HT_NEW(ii,jj)) / ((Lx(ii-1,jj)+Lx(ii,jj))/2.)
                s2 = (HT_NEW(ii-1,jj+1) - HT_NEW(ii,jj+1)) / ((Lx(ii-1,jj+1)+Lx(ii,jj+1))/2.)
            ENDIF

            s3 = (HT_NEW(ii,jj) - HT_NEW(ii+1,jj)) / ((Lx(ii,jj)+Lx(ii+1,jj))/2.)
            s4 = (HT_NEW(ii,jj+1) - HT_NEW(ii+1,jj+1)) / ((Lx(ii,jj+1)+Lx(ii+1,jj+1))/2.)

            IF(ii==1) then
                Sxv(ii,jj) = (s3 + s4) / 2.
            ELSE
                Sxv(ii,jj) = (s1 + s2 + s3 + s4) / 4.
            ENDIF
            sfv(ii,jj) = SQRT(sxv(ii,jj)**2 + syv(ii,jj)**2)
        END DO; END DO

! Diffusion coefficient / Conveyance b/w longitudinally neighboring cells
        DO ii=1,Nx-1; DO jj=1,Ny
            hmid = (H_NEW(ii,jj)*Lx(ii+1,jj) + H_NEW(ii+1,jj)*Lx(ii,jj)) &
                    / (Lx(ii,jj)+Lx(ii+1,jj))

            IF(sfh(ii,jj)/=0) then
                Diffx(ii,jj) = phi_e * hmid**(5./3.) / (mann * SQRT(sfh(ii,jj)))
            else
                Diffx(ii,jj) = 0
            endif
            csx(ii,jj) = 2. * Diffx(ii,jj) / ((Lx(ii,jj)+Lx(ii+1,jj)) * Lx(ii,jj))
        END DO; END DO
```

```
! Diffusion coefficient / Conveyance b/w laterally neighboring cells
        DO ii=1,Nx-1; DO jj=1,Ny-1
            hmid = (H_NEW(ii,jj) + H_NEW(ii,jj+1)) / 2.

            IF(sfv(ii,jj)/=0) then
                Diffy(ii,jj) = phi_e * hmid**(5./3.) / (mann * SQRT(sfv(ii,jj)))
            else
                Diffy(ii,jj) = 0
            endif
            csy(ii,jj) = Diffy(ii,jj) / dy ** 2
        END DO; END DO

END SUBROUTINE

!*****************************************************************************************


!       Last change:   JJ     19 Aug 2007       2:57 pm
SUBROUTINE post(ti, tf)

        USE resource

        REAL, INTENT(IN) :: ti, tf

! Display the total run time on the screen
        WRITE(*,*) ''
        WRITE(*,*) ' RUNTIME : ', INT((tf-ti)/60.), ' min ',INT(((tf-ti)/60.-INT((tf-ti)/60.))*60), ' sec'

! Display the time when the last run finished
        call date_and_time(date,time)
        WRITE(*,*) ''
        WRITE(*,*) ' Time Finished : ',time(1:2),':',time(3:4),':',time(5:6),'     ',date(1:4),'/',date(5:6),'/',date(7:8)

! Surface elevation (Array)
        do ii=1, Nx; do jj=1,Ny
            WRITE(1000,800) x(ii,jj), y(ii,jj), z(ii,jj)
!            WRITE(1000,100) ii*dx, jj*dy, z(ii,jj)
        end DO; end do

! Surface elevation (Scalar)
        do ii=1,Nx
            WRITE(1100,100) (z(ii,jj), jj=1,Ny)
        end do
! Surface elevation (Scalar)
        do ii=0,Nx
            WRITE(1150,100) (zb(ii,jj), jj=1,Ny)
        end do

! Surface elevation (Edges of cell, Array)
        do ii=0, Nx; do jj=0,Ny
```

```
            WRITE(1200,800) xb(ii,jj), yb(ii,jj), zb(ii,jj)
        end DO; end do

! Grid mesh -x
        do ii=1, Nx
            WRITE(1300,1000) (ii, jj=1,Ny)
        end DO

! Grid mesh -y
        do ii=1, Nx
            WRITE(1400,1000) (jj, jj=1,Ny)
        end DO

! Longitudinal length of each grid cell (Scalar)
        do ii=1,Nx
           WRITE(1500,100) (Lx(ii,jj), jj=1,Ny)
           WRITE(1600,100) (Lb(ii,jj), jj=0,Ny)
        end do

! conveyances
        do ii=1,Nx-1
           WRITE(900,100) (csx(ii,jj), jj=1,Ny-1)
           WRITE(910,100) (csy(ii,jj), jj=1,Ny-1)
        end do

! Water depth and Total head (Steady, Array)
        DO ii=1,Nx; do jj=1,Ny
            WRITE(100,800) x(ii,jj), y(ii,jj), 304.8*H_NEW(ii,jj)
            WRITE(101,800) x(ii,jj), y(ii,jj), HT_NEW(ii,jj)
!            WRITE(100,900) ii, jj, H_NEW(ii,jj)
!            WRITE(101,900) ii, jj, HT_NEW(ii,jj)
        END DO; END do

! Water depth and total head (Steady, Scalar)
        DO ii=1,Nx
            WRITE(160,100) (304.8*H_new(ii,jj), jj=1,Ny)
            WRITE(151,100) (Ht_new(ii,jj), jj=1,Ny)
        END DO

! Unit flow rate (Steady, Array)
        DO ii=1,Nx-1,2
            do jj=1,Ny,3
                WRITE(500,900) ii, jj, qx(ii,jj)
                WRITE(600,900) ii, jj, qy(ii,jj)
            END DO
            IF(jj.eq.Ny) WRITE(500,900) ii, Ny, qx(ii,Ny)
            IF(jj.eq.Ny) WRITE(600,900) ii, Ny, qy(ii,Ny-1)
        END do
```

```
! Variables related to geometric characteristics
        WRITE(6000,400) 'DTM File Name :                    ', filename
        WRITE(6000,300) 'Domain Size :          (L,W)     = ', INT(L_cum), INT(width)
        WRITE(6000,300) 'Number of cells :     (Nx,Ny) = ',Nx, Ny
        WRITE(6000,500) 'Intervals :          (dx,dy,dt) = ',dx, dy, dt
        WRITE(6000,400) 'Curvature Shape :                  ',curve_pattern
        WRITE(6000,*) ''
        WRITE(6000,*) 'Run Time : ', INT((tf-ti)/60.), ' min ',INT(((tf-ti)/60.-INT((tf-ti)/60.))*60), ' sec'
        WRITE(6000,*) ''
        WRITE(6000,600) 'Maximum Surface Slope :    Sx = ',MAXVAL(Sox), 'Sy =',MAXVAL(Soy)
        WRITE(6000,600) 'Minimum Surface Slope :    Sx = ',MINVAL(Sox), 'Sy =',MINVAL(Soy)
        WRITE(6000,*) ''
        WRITE(6000,500) 'Rainfall Intensity :          ',ri*12*3600
        WRITE(6000,600) 'Maximum unit discharge[cm2/s]: ',MAXVAL(SQRT(qx**2+qy**2))*30.48**2
        WRITE(6000,600) 'Maximum ponding depth[mm] :     h = ',MAXVAL(H_new)*304.8
        WRITE(6000,550) 'Time to initiate flow =           ',t_ini, '[sec]'
        WRITE(6000,550) 'Time to reach Steady State =    ',t_accum/60.,'[min]'
        WRITE(6000,*) ''
        WRITE(6000,*) ''

        WRITE(6000,*) ' Radius of curvature at stations'
        do ii=0,inum
           if (shp(ii).eq.'L') then
              WRITE(6000,700) 'N/A'
           else
              WRITE(6000,200) rc(ii)
           end if
        end do
        WRITE(6000,*) ''
        WRITE(6000,*) ''
         WRITE(6000,*) ' Direction of the roadway w.r.t. the horizontal axis [deg]'
        do ii=1,Nx-1, INT(Nx/inum)
          WRITE(6000,200) gamma(ii)* 180 / pi
        end do
        WRITE(6000,*) ''
         WRITE(6000,*) ' Downstream B.C. type '
        do jj=1,Ny
          WRITE(6000,*) REAL(jj)/Ny, DNBC(jj)
        end do

100      FORMAT(1x,1000f10.5)
200      FORMAT(1x,f20.2)
300     FORMAT(1x,a30,3I5)
400     FORMAT(1x,a30,a20)
500     FORMAT(1x,a30,3f7.3)
550     FORMAT(1x,a30,f7.3,a5)
600     FORMAT(1x,a30,e10.4,a6,e10.4)
700      FORMAT(1x,a20)
800     FORMAT(3f20.5)
900     FORMAT(2I10,f10.5)
```

```
1000      FORMAT(10000I10)

END SUBROUTINE

!       Last change:   JJ    29 Apr 2007      1:40 pm
subroutine output_files()

          USE resource
          REAL :: dummy

! ********************************************************************
!   Definition
! ********************************************************************
!
!   Unsteady   :   Writes output repeatedly at specified time steps
!
!   Steady     :   Writes only the Steady State result
!                  (Output of the very last time step)
!
!   Array      :   Writes output in a (x,y,value) format
!                  ex) 1.0      1.0       0.01
!                      2.0      1.5       0.02
!
!   Scalar     :   Writes output for spreadsheet display
!                  ex) 0.01     0.02     0.02     0.03
!                      0.02     0.02     0.03     0.03
!                      0.02     0.03     0.03     0.04
!
! ********************************************************************
!
! Surface elevation (Center of cell, Array)
          OPEN (UNIT=1000, FILE='zb_vec.txt', STATUS='replace', IOSTAT=IERROR)
! Surface elevation (Center of cell, Scalar)
          OPEN (UNIT=1100, FILE='z_spa.txt', STATUS='replace', IOSTAT=IERROR)
! Surface elevation (Corner of cell, Scalar)
          OPEN (UNIT=1150, FILE='zb_spa.txt', STATUS='replace', IOSTAT=IERROR)
! Surface elevation (Edges of cell, Array)
          OPEN (UNIT=1200, FILE='zb_blocks.txt', STATUS='replace', IOSTAT=IERROR)
! Grid mesh - x
          OPEN (UNIT=1300, FILE='xi.txt', STATUS='replace', IOSTAT=IERROR)
! Grid mesh - y
          OPEN (UNIT=1400, FILE='yi.txt', STATUS='replace', IOSTAT=IERROR)
! Longitudinal length of each grid cell (Scalar)
          OPEN (UNIT=1500, FILE='Lx.txt', STATUS='replace', IOSTAT=IERROR)
! Longitudinal length of each cell boundary (Scalar)
          OPEN (UNIT=1600, FILE='Lb.txt', STATUS='replace', IOSTAT=IERROR)
! Variables related to geometric characteristics
            OPEN (UNIT=6000, FILE='control_variables.txt', STATUS='replace', IOSTAT=IERROR1)
! Water depth (Unsteady, Array)
          OPEN(UNIT=100, FILE='H_vec.txt', STATUS='replace', IOSTAT=IERROR)
```

```
! Total head (Unsteady, Array)
        OPEN(UNIT=101, FILE='Ht_vec.txt', STATUS='replace', IOSTAT=IERROR)
! Water depth (Unsteady, Scalar)
        OPEN(UNIT=150, FILE='h_spa.txt', STATUS='replace', IOSTAT=IERROR)
! Water depth (Stedy, Scalar)
        OPEN(UNIT=160, FILE='h_spaSS.txt', STATUS='replace', IOSTAT=IERROR)
! Total head (Unsteady, Scalar)
        OPEN(UNIT=151, FILE='HT_spaSS.txt', STATUS='replace', IOSTAT=IERROR)
! Water depth at the center along the roadway (Unsteady, Scalar)
        OPEN(UNIT=200, FILE='h(x,t).txt', STATUS='replace', IOSTAT=IERROR)
! L2, L infinity norms for convergence of the CG solver
        OPEN(UNIT=400, FILE='norm_cg.txt', STATUS='replace', IOSTAT=IERROR)
! Norms for convergence of nonlinearity iteration
        OPEN(UNIT=425, FILE='norm_pic.txt', STATUS='replace', IOSTAT=IERROR)
! Norms for steady state
        OPEN(UNIT=450, FILE='norm_SS.txt', STATUS='replace', IOSTAT=IERROR)
! Longitudinal component of unit flow rate (Steady, Array)
        OPEN(UNIT=500, FILE='qx.txt', STATUS='replace', IOSTAT=IERROR)
! Transverse component of unit flow rate (Steady, Array)
        OPEN(UNIT=600, FILE='qy.txt', STATUS='replace', IOSTAT=IERROR)
! Actual time at each time step (seconds)
        OPEN(UNIT=800, FILE='time.txt', STATUS='replace', IOSTAT=IERROR)
! X-conveyance
        OPEN(UNIT=900, FILE='csx_SS.txt', STATUS='replace', IOSTAT=IERROR)
! Y-conveyance
        OPEN(UNIT=910, FILE='csy_SS.txt', STATUS='replace', IOSTAT=IERROR)

        if (IERROR.NE.0) then
            WRITE(*,*) " Error occured in creating output files... Program terminated."
          call EXIT(0)        ! Terminate program
        END if
end subroutine
```

C.2 OUTPUT FILES

C.2.1 Computation Related Files

control_variables.txt: Source file name, control variables, summary of result
norm_SS.txt: Histories of L_2 and L_∞ norms in the conjugate gradient solver iteration
norm_pic.txt: History of L_2 and L_∞ norms in the nonlinear iteration
norm_cg.txt: History of L_2 and L_∞ norms before the computation reaches steady state

C.2.2 Geometry Related Files

geometric_data.txt: Center of road, radius of curvature, direction, and origin of curvature
zb_vec.txt: Elevation of surface at the edge of cells; vector (x,y,z) format
zb_spa.txt: Elevation of surface at the edge of cells; matrix format
Lb.txt: Horizontal length between the centers of neighboring cells
z_vec.txt: Elevation of surface at the center of cells; vector (x,y,z) format
z_spa.txt: Elevation of surface at the center of cells; matrix format
Lx.txt: Length of cells

C.2.3 Result Files

h_spaSS.txt: Steady state water depth at each cell, matrix format
H_vec.txt: Steady state water depth at each cell, vector format
h(x,t).txt: Time-series water depth of center of road
HT_spaSS.txt: Steady state total head (H=z+h), matrix format
HT_vec.txt: Steady state total head, vector format
csx_SS.txt: Conveyances in the X-direction at Steady state
csy_SS.txt: Conveyances in the Y-direction at Steady state
qx.txt: Unit flow rate in the X-direction at Steady state
qy.txt: Unit flow rate in the Y-direction at Steady state

C.2.4 Example

File name: *control_variables.txt*

```
DTM File Name :      super1r.dat
Domain Size :        (L,W)      =      100      44
Number of cells :      (Nx,Ny) =        52      65
Intervals :          (dx,dy,dt) =      1.923    0.985    0.400
Curvature Shape :    clockwise

Run Time :    6    min    51    sec

Maximum Surface Slope :    Sx = 0.2820E-01    Sy =0.3744E-01
Minimum Surface Slope :    Sx = 0.4300E-04    Sy =-.1943E-01

Rainfall Intensity :              10.000
Maximum unit discharge[cm2/s]:0.1793E+02
Maximum ponding depth[mm] :      0.6668E+01
Time to initiate flow =                  0.004[sec]
Time to reach Steady State =        2.507[min]


 Radius of curvature at stations
                              N/A
                              N/A
                              N/A
                              N/A
                              N/A


 Direction of the roadway w.r.t. the horizontal axis [deg]
                           93.20
                           93.20
                           93.19
                           93.20
```

References

AASHTO (1999). *Highway Drainage Guidelines [Metric Edition],* American Association of State Highway and Transportation Officials.

AASHTO (2004). *Geometric Design of Highways and Streets*, American Association of State Highway and Transportation Officials.

Agiralioglu, N. (1981). "Water Routing on Diverging-Converging Watersheds." Journal of the Hydraulics Division-ASCE, 107(8): p. 1003-1017.

Akan, A. O., G. C. Schafran, et al. (2000). "Modeling storm-water runoff quantity and quality from marine drydocks." Journal of Environmental Engineering-Asce **126**(1): 5-11.

Anderson, D.A., Huebner, R.S., Reed, J.R., Warner, J.C., and Henry, J.J. (1998). "Improved Surface Drainage of Pavements." NCHRP Web DOC 16.

ASCE (1982). *Gravity sanitary sewer design and construction.* ASCE manual of engineering practice, No. 60, ASCE, New York.

ASTM (2006). *Standard Test Method for Measuring Pavement Macrotexture Depth Using a Volumetric Technique Designation: E 965 – 96,* ASTM International.

Charbeneau, R.J., Jeong, J., Reeder, E., and Chan, W.S. (2007). "Physical modeling of sheet flow on highway pavement surfaces." Presented at the 32nd Congress of IAHR, Venice, Italy, July 1-6, 2007, International Association of Hydraulic Engineering and Research.

Chow, V.T. (1959). *Open Channel Hydraulics*, McGraw-Hill, New York.

Chow, V. T., Maidment, D. R., Mays, L. W. (1988). *Applied Hydrology*, McGraw-Hill, New York.

Cooley, R. L (1992). *A Modular Finite-Element Model (MODFE) for Areal and Axisymmetric Groundwater Flow Problems, Part 2: Derivation of Finite-Element Equations and Comparisons with Analytical Solutions*. USGS Book 6, Chapter A4.

Cristina, C. M. and J. J. Sansalone (2003). "Kinematic wave model of urban pavement rainfall-runoff subject to traffic loadings." Journal of Environmental Engineering-Asce **129**(7): 629-636.

Feng, K. and F.J. Molz (1997). "A 2-D, diffusion-based, wetland flow model." Journal of Hydrology, 196(1-4): p. 230-250.

Ferziger, J.H. and Peric, M. (2002). *Computational Methods for Fluid Dynamics*. Springer, Berlin.

Johnson, F.L. and F. F.M. Chang (1984) *Drainage of Highway Pavements (HEC-12)*, Federal Highway Administration, McLean, Virginia.

Brown, S.A., Stein, S.M., Warner, J.C. (2001), *Urban Drainage Design Manual (HEC-22)*, Federal Highway Administration, McLean, Virginia.

Govindaraju, R. S., S. E. Jones, et al. (1988a). “On the Diffusion Wave Model for Overland-Flow .1. Solution for Steep Slopes.” Water Resources Research 24(5): 734-744.

Govindaraju, R. S., S. E. Jones, et al. (1988b). “On the Diffusion Wave Model for Overland-Flow .2. Steady-State Analysis.” Water Resources Research 24(5): 745-754.

Grayson, R. B., Moore, I. D., and McMahon, T. A. (1992). “Physically Based Hydrologic Modeling 2. Is the Concept Realistic?” Water Resourc. Res. 26, 2659–2666.

Henderson, F. M. (1966). *Open Channel Flow*. MacMillan Publishing Co., Inc., NY, NY.

Henderson, F.M. and Wooding, R.A. (1964). “Overland flow and groundwater flow from a steady rainfall of finite duration.” J. Geophysical Research, 69(8), 1531-1540.

Hirsch, Ch (1988). *Numerical Computation of Internal & External Flows. Volume 1*, John Wiley & Sons Ltd..

Hodges, B. R. (2004). “Accuracy order of Crank-Nicolson discretization for hydrostatic free-surface flow.” Journal of Engineering Mechanics-Asce 130(8): 904-910.

Hromadka, T.V. and C.C. Yen (1986). “A Diffusion Hydrodynamic Model (DHM).” Advances in Water Resources, 9(3): p. 118-170.

Huebner, R. S., J. R. Reed, et al. (1986). "Criteria for Predicting Hydroplaning Potential." Journal of Transportation Engineering-Asce **112**(5): 549-553.

Iwagaki, Y. (1955). “Fundamental studies of the runoff analysis by characteristics.” Kyoto Univ. Disaster Prevention Res. Inst., Japan, Bull., 10, 1-25.

Jain, M.K. and V.P. Singh (2005). “DEM-based modeling of surface runoff using diffusion wave equation.” Journal of Hydrology, 302(1-4): p. 107-126.

Jeong, J. and Charbeneau, Randall J. (2007). “Numerical Simulation of Stormwater runoff from Highway Pavements through Superelevation Transitions.” Presented at the 32nd Congress of IAHR, Venice, Italy, July 1-6, 2007, International Association of Hydraulic Engineering and Research.

Kazezyilmaz-Alhan, C. M. and M. A. Medina (2007). "Kinematic and diffusion waves: Analytical and numerical solutions to overland and channel flow." Journal of Hydraulic Engineering-Asce 133(2): 217-228.

Keulegan, G.H (1944). "Spatially variable discharge over a sloping plane." Trans. Am. Geophys. Union, 25, 956-959.

Lal, A.M.W. (1998). "Weighted implicit finite-volume model for overland flow." Journal of Hydraulic Engineering-ASCE, 124(9): p. 941-950.

Liggett, J.A. and D.A. Woolhiser (1967). "Difference Solutions of the Shallow-Water Equation." Journal of the Engineering Mechanics Division, EM2: p. 39-71.

Lighthill, M. J. and G. B. Whitham (1955a). "On kinematic waves, 1. Flood movement in long rivers." Proc. Roy. Soc. London **A**(229): 281-316.

Lighthill, M. J. and G. B. Whitham (1955b). "On kinematic waves, II. A theory of traffic flow on long crowded roads." Roy. soc. London **A**(229): 317-345.

Ponce, V. M. (1990). "Generalized Diffusion Wave-Equation with Inertial Effects." Water Resources Research **26**(5): 1099-1101.

Reeder, Emily K. (2007). *Effects of longitudinal slope, roughness, and rainfall on drainage parameters*. Thesis. The University of Texas at Austin.

Sherman, L. K. (1932). "Streamflow from rainfall by the unit-hydrograph method." Eng. News Rec., 108, 501–505

Schneider, Lauren (2006). *Physical models of rainfall runoff for superelevation transition*. Thesis. The University of Texas at Austin.

Singh, V. P. and N. Agiralioglu (1981a). "Diverging Overland-Flow .1. Analytical Solutions." Nordic Hydrology, 12(2): 81-98.

Singh, V.P. and N. Agiralioglu (1981b), "Diverging Overland-Flow .2. Application to Natural Watersheds." Nordic Hydrology, **12**(2): p. 99-110.

Singh, V.P. and V. Aravamuthan (1996). "Errors of kinematic-wave and diffusion-wave approximations for steady-state overland flows." Catena, 27(3-4): p. 209-227.

Tisdale, T. S. and J. M. H. L. Yu (1999). Kinematic wave analysis of sheet flow using topography fitted coordinates." Journal of Hydrologic Engineering 4(4).

Tsai, T. L. and J. C. Yang (2005). "Kinematic wave modeling of overland flow using characteristics method with cubic-spline interpolation." Advances in Water Resources 28(7): 661-670.

Vieira, J.H.D. (1983). “Conditions Governing the Use of Approximations for the Saint-Venant Equations for Shallow Surface-Water Flow.” Journal of Hydrology, 60(1-4): p. 43-58.

Villard, Julien (2005). *Minimum longitudinal grade at zero cross slope in superelevation transition.* Thesis. The University of Texas at Austin.

Wooding, R. A. (1966). “A hydraulic model for the catchment-stream problem: 3. Comparison with runoff observations.” Journal of Hydrology, V4, 21-37

Woolhiser, D.A. (1969). “Overland flow on a converging surface.” Transactions of the ASAE, V12, No 4, 460-462

Woolhiser, D.A. and J.A. Liggett (1967). “Unsteady, one-dimensional flow over a plane – the rising hydrograph.” Water Resources Research, 3, 753-771.

Yu, Y.S. and J.S. McNown (1964). “Runoff from impervious surfaces”, J. Hydraulics Research, 2(1), 3-24

Zhang, W.H. and Cundy, T.W. (1989). “Modeling of Two-Dimensional Overland-Flow.” Water Resources Research, 25(9): p. 2019-2035.

Zhang, W.H. and Montgomery, D.R. (1994). “Digital Elevation Model Grid Size, Landscape Representation, and Hydrologic Simulations.” Water Resources Research, 30(4): p. 1019-1028.

Vita

Jaehak Jeong was born in Chuncheon, South Korea on March 23, 1974. He is the son of Kisup Jeong and Keumja Park. Jaehak completed his work of high school at Chuncheon High School in Chuncheon, in 1993. Upon graduation from the high school he attended Korea University in Seoul earning the degree Bachelor of Science in Civil and Environmental Engineering in February 2000. He served as a teaching assistant at the Engineering School of Korean Army for two years from May 1995 to July 1997. After completing his undergraduate degree, he began working full time as a civil engineer at Hyundai Development Inc. from January 2000. Jaehak entered the Graduate School of The University of Texas at Austin in September 2002, earned his Master of Science degree in December 2004, and continued his study for Ph.D. at the same institution.

Permanent address: 227 Gangchon 2-Ri Namsan-Myeon Chuncheon, Kangwon
200-912 South Korea

This dissertation was typed by the author.