Characterizing and Understanding the Effects of Spatial Resolution in Urban Hydrologic Simulations

A Dissertation Presented
by
Indrani Ghosh
to
The Department of Civil and Environmental Engineering
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in the field of
Environmental Engineering

Northeastern University
Boston, Massachusetts

(Dissertation completed in August 2010)
Abstract

Model subdivision is used to capture spatial heterogeneity in input parameters and it is well-established that spatial resolution affects model output. Although previous research has observed scale effects in urban hydrology, there is no general consensus about it or the underlying mechanism(s). The objective of this study was to investigate the effects of spatial resolution on model predictions in an urban catchment, and to understand the mechanism(s) responsible for the scale effect. The study area is the Faneuil Brook sub-basin of the lower Charles River watershed in Boston. The general approach used in this study is to develop models at various spatial resolutions (from 4 to 616 subcatchments), perform simulations and compare the predictions of total outflow volume and peak flow. Models were developed based on actual drainage networks, and artificial ones generated based on a fractal algorithm using the program Artificial Network Generator (ANGel), which was written as part of this research. Simulations were performed using the EPA Storm Water Management Model (SWMM) and model output was compared for 90 different storms.

There was very little difference in the total annual outflow volumes predicted by the different resolutions. However, peak flows showed a dual scale effect based on storm characteristics differentiated by total rainfall depth. For the larger storms, model aggregation significantly reduced peak flows, which can be explained by differences in infiltration and runoff. This effect was attributed mainly to the spatial distribution of the soil saturated hydraulic conductivity and the length of overland flow. For the smaller storms, aggregation significantly increased peak flows, which can be explained by the combined effects of overland flow and conduit routing. The results were consistent using actual and artificial networks, and also for storms in other water years. This study illustrates that a scale effect can be introduced by different processes, which can go in
different directions (i.e. increase or decrease peak flows) and depend on the storm characteristics. This study also illustrates that the peak flows and the effects of spatial resolution are comparable using actual and artificial networks, which implied that artificial sewer networks can be a useful tool for spatial scaling analysis.
Acknowledgements

The completion of my PhD dissertation has been a long journey and looking back I truly value all the experience that I have gained during the course of my doctoral studies here at Northeastern University. I am grateful for everything that I have received throughout these years and this is a great opportunity for me to thank everyone who made this thesis possible.

I would like to express my deep and sincere gratitude to my supervisor Professor Ferdinand L. Hellweger for mentoring me in the area of urban hydrology. His expertise, guidance and critique have been extremely valuable contributions to my research and greatly enhanced my PhD experience. I would like to thank Professor Vladimir Novotny, Professor Akram Alshawabkeh and Dr. David Bedoya for serving on my committee, and for their insightful questions and advice on my research. It is also my pleasure to thank Professor Thomas C. Sheahan and Professor Irvine Wei for their encouragement during the course of my doctoral studies. I gratefully acknowledge the Department of Civil and Environmental Engineering for providing me the financial support as a Teaching Assistant during the five years of my PhD program.

I would like to express my sincere thanks to Robert C. James of Computational Hydraulics International who provided the licensed version of the software program PCSWMM free of cost, and was always willing to resolve any issue related to the implementation of the software. I am grateful to Phillip J. Zarriello of the United States Geological Survey (USGS), Amy Schofield, and Vincent Mannering of Boston Water and Sewer Commission (BWSC) for providing me the necessary data during the initial stages of my thesis. I would also like to make a special mention of the online Stormwater Management Model (SWMM) Users forum, and its numerous users who have answered several of my questions related to model implementation.
The time I spent at Northeastern was especially enjoyable due to the many friends and colleagues who made my whole PhD experience an enriching one. Be it sharing research agonies over coffee or planning fun field-trips at the Graduate Environmental and Geo-Environmental Society (GEGES), these will always remain as some of my most cherished memories of graduate student life. I would like to thank my friends Carla, Nehreen, Miriam, Vanni, Arash and all other fellow graduate students in the Civil and Environmental Engineering department for the wonderful time that I spent with them.

Finally, I dedicate my thesis to my family: my husband Manidip, my parents in India and my sister Debyani. They have been the permanent pillars of my life, on whom I can lean upon for support in times of stress and disappointment. Without their love, encouragement and understanding, it would not have been possible for me to finish this work and achieve my pursuits.

Indrani Ghosh
Northeastern University
August 2010
# Table of Contents

**Table of Contents**

- Table of Contents .......................................................................................... vi
- List of Figures ..................................................................................................... ix
- List of Tables ...................................................................................................... xvi

**Chapter 1. Introduction** .................................................................................. 1

1.1 Modeling Urban Hydrology ........................................................................... 1
1.2 Model Subdivision and its Effects .................................................................. 2
1.3 Problem Statement ......................................................................................... 4
1.4 Research Objectives and Approach .............................................................. 6
1.5 Thesis Organization ....................................................................................... 7
1.6 References ...................................................................................................... 9

**Chapter 2. Literature Review** .......................................................................... 10

2.1 Rural Hydrology ............................................................................................ 10
2.2 Urban Hydrology ........................................................................................... 17
2.3 Summary and Open Questions ...................................................................... 21
2.4 References ..................................................................................................... 25

**Chapter 3. Approach and Detailed Methodology** ............................................. 28

3.1 Study Area ..................................................................................................... 29
3.2 Input Data Preparation and Modifications to Original Model ..................... 30
3.3 Generating Artificial Sewer Networks ............................................................ 32
3.4 Methodology (Fractal Geometry) ................................................................... 34
3.5 User Input Options ......................................................................................... 37
  3.5.1 Starting Network ...................................................................................... 39
  3.5.2 Existing Drainage Areas ........................................................................ 39
  3.5.3 Combining Pipes in Drainage Network .................................................. 40
  3.5.4 Splitting Pipes in Drainage Network ....................................................... 42
  3.5.5 Fractal Type ............................................................................................ 44
  3.5.6 Number of Generations to Add ............................................................... 44
  3.5.7 Advanced Options .................................................................................. 45
  3.5.8 Input Options for SWMM ........................................................................ 47
Chapter 4. Results of Scale Effects

4.1 Peak Flows for All Storms
   4.1.1 Storms in Water Year 2000
   4.1.2 Storms in Water Year 1999 and 2001

4.2 Scale Effect in Peak Flows for Large Storms
   4.2.1 Large Storms in Water Year 2000
   4.2.2 Large Storms in Water Year 1999 and 2001
   4.2.3 Effect of Antecedent Moisture in Large Storms

4.3 Scale Effect in Peak Flows for Small Storms
   4.3.1 Small Storms in Water Year 2000
   4.3.2 Small Storms in Water Year 1999 and 2001

4.4 References

Chapter 5. Analysis of Scale Effects

5.1 Analysis of Scale Effect in Large Storms
   5.1.1 Water Balance Parameters for Large Storms
   5.1.2 Effect of Spatial Distribution of Parameters
   5.1.3 Effect of Distributed Property $K_s$
   5.1.4 Effect of Distributed Property Length of Overland Flow (LOF)
   5.1.5 Combined Effects of $K_s$ and LOF at Different Rainfall Intensities

5.2 Analysis of Scale Effect in Small Storms
   5.2.1 Water Balance Parameters for Small Storms
   5.2.2 Effect of Overland Flow Routing
   5.2.3 Effect of Conduit Routing
   5.2.4 Combined Effect of Overland Flow and Conduit Routing

5.3 References
Chapter 6. Comparison of Actual and Artificial Networks..........................123
  6.1 Comparison of Peak Flows....................................................................123
  6.2 Comparison of Scale Effect......................................................................130
    6.2.1 Larger Storms..................................................................................130
    6.2.2 Smaller Storms................................................................................132

Chapter 7. Summary and Recommendations for Further Studies...............134
  7.1 Summary.................................................................................................134
  7.2 Recommendations for Further Studies....................................................137

Appendix 1. ANGel Tutorial...........................................................................139
Appendix 2. Computer Code for ANGel.........................................................147
List of Figures

Chapter 1. Introduction

Figure 1.1 Illustration of the effects of non-linearity and spatial resolution on model results.
(a) Rational method (linear model) produces same peak flow at different resolutions;
(b) curve number method produces different runoff at different resolutions;
(c) Comparison of peak flow per unit area for the two different resolutions using spatially variable Cs;
(d) Comparison of runoff depths for the two different resolutions using spatially variable CNs……………………………………………………………………………..4
Figure 1.2 Flow hydrographs predicted by applying SWMM to an urban catchment at two different spatial resolutions in a study conducted by Zaghloul (1981)…………….5

Chapter 3. Approach and Detailed Methodology

Figure 3.1. Faneuil Brook sub-basin of the lower Charles River watershed, showing the actual sewer network and the pond (Zarriello and Barlow, 2002).......................29
Figure 3.2. Construction of a “Tokunaga” fractal tree (from Turcotte and Newman, 1996)…35
Figure 3.3 User Input Window of the Artificial Network Generator (ANGel), version 1.2….38
Figure 3.4 (a) A fractal network with user-specified starting network,
(b) A fractal network with existing shapefile as starting network.........................39
Figure 3.5. Output from ANGel using
(a) existing drainage network with existing drainage areas, and
(b) existing drainage network with artificial drainage areas...............................40
Figure 3.6. (a) Original network with shorter pipes
(b) Modified network with shorter pipes merged. Note the shorter pipe segments in (a) have been merged to form longer pipes in (b).................................41
Figure 3.7. (a) Coarser resolution network with longer pipes
(b) Finer resolution network with smaller pipes, generated using the “Split Distance” option in ANGel.................................................................42
Figure 3.8. (a) Original network with smaller pipes
(b) Original pipes that were merged to form longer pipes, and were subsequently split into shorter pipes of equal lengths.................................................43
Figure 3.9. (a) A fractal network generated by adding zero generations to an existing shapefile.
(b) A fractal network generated by adding four generations to a starting network,
(c) A complex fractal network generated by adding five generations to a starting network.

Figure 3.10. A fractal network
(a) before, and
(b) after clipping to the extents of a catchment area.

Figure 3.11. (a) A regular fractal network.
(b) An irregular network with 30 percent irregularity.

Figure 3.12. Input data for percent imperviousness as raster grids (shown as graded colors)
(a) higher resolution than resolution of subcatchment areas;
(b) lower resolution than resolution of subcatchment areas.

Figure 3.13. Program flow in the initial stage when existing drainage network is used.

Figure 3.14. Program flow when existing drainage areas are used for analysis or when they are generated artificially.

Figure 3.15. Program flow to add generations to the starting network.

Figure 3.16. Program flow to generate SWMM attributes for the nodes, pipes and areas shapefiles.

Figure 3.17 (a) Input raster dataset for subcatchment slope (in %) derived from initial model set up by Zariello and Barlow (2002);
(b) raster grids for slope overlaid on drainage areas (corresponding to an artificial network) to calculate slope for subcatchments using the Zonal Statistics function in ANGel.

Figure 3.18 (a) Input raster dataset for saturated hydraulic conductivity derived from SSURGO data for Norfolk-Suffolk and Middlesex counties;
(b) raster grids for saturated hydraulic conductivity overlaid on drainage areas (corresponding to actual network at a coarse resolution) to calculate $K_s$ for subcatchments using the Zonal Statistics function in ANGel.
Chapter 4. Results of Scale Effects

Figure 4.1. GIS representations of actual and artificial networks (pipes and nodes) with their corresponding drainage areas.
(a) Actual L1 with 4 areas,
(b) Actual L2 with 19 areas;
(c) Actual L3 with 401 areas;
(d) Artificial L1 with 5 areas;
(e) Artificial L2 with 21 areas; and
(f) Artificial L3 with 616 areas…………………………………………………………………77

Figure 4.2. Actual L2 benchmarked with original model for storm on Dec 6…………………78

Figure 4.3. Percent difference in peak flows with respect to
(a) Actual L3 and
(b) Artificial L3 plotted against total rainfall depth for 50 storms of water year
2000……………………………………………………………………………………………….79

Figure 4.4 Percent difference in peak flows with respect to
(a) Actual L3 and
(b) Artificial L3 plotted against total rainfall depth for 20 storms of water year
2000.
Percent difference in peak flows with respect to
(c) Actual L3 and
(d) Artificial L3 plotted against total rainfall depth for 20 storms of water year
2001. …………………………………………………………………………………………….81

Figure 4.5. (a) Outflow hydrograph and hyetograph,
(b) Runoff hydrograph for July 27, 2000 storm compared at resolutions Actual L1, Actual L2, Actual L3……………………………………………………………………………………………….83

Figure 4.6. (a) Outflow hydrograph and hyetograph,
(b) Runoff hydrograph for June 2 storm compared at resolutions Actual L1, Actual L2, Actual L3………………………………………………………………………………………………………….84

Figure 4.7. (a) Outflow hydrograph;
(b) system losses;
(c) surface runoff hydrograph for storm on October 8, 1998…………………..86

Figure 4.8. (a) Outflow hydrograph;
(b) system losses;
(c) surface runoff hydrograph for storm on March 30, 2001……………………………………………………86
Figure 4.9. Percent difference in peak flows predicted by Actual L1 with respect to Actual L3 plotted against total rainfall depth and antecedent moisture conditions for 14 largest storms of water year 2000…………………………………………………………. 88

Figure 4.10. Percent difference in peak flows predicted by Actual L1 with respect to Actual L3 plotted against total rainfall depth. Peak flows predicted by Actual L1 for storms on April 22 and July 27 run with dry antecedent conditions are shown as hollow markers…………………………………………………………………………….. 89

Figure 4.11. (a) Outflow hydrograph and hyetograph, (b) runoff hydrograph for July 22, 2000 storm compared at resolutions Actual L1, Actual L2, Actual L3………………………………………………………………………………… 90

Figure 4.12. (a) Outflow hydrograph, (b) runoff hydrograph and hyetograph, and (c) water balance parameters for June 2 storm compared at resolutions Actual L1, Actual L2, Actual L3……………………………………………………………………………… 91

Figure 4.13. (a) Outflow hydrograph; (b) system losses; (c) surface runoff hydrograph for storm on July 6, 1999………………………………………………………………………………………… 93

Figure 4.14 (a) Outflow hydrograph; (b) system losses; (c) surface runoff hydrograph for storm on September 13, 2001………………………………………………………………………………………… 93

Chapter 5. Analysis of Scale Effects

Figure 5.1 Water balance parameters and peak flow rates compared for July 27 storm at resolutions Actual L1, Actual L2, Actual L3. \( Q_{\text{shed}} \): total runoff (in.); INFIL: infiltration (in.); E: evaporation (in); ETU: upper zone evapotranspiration (in.); ETD: lower zone evapotranspiration (in.); \( Q_{p,\text{run}} \): peak runoff (cfs); \( Q_{p,\text{out}} \): peak outflow (cfs)………………95

Figure 5.2 Water balance parameters and peak flow rates compared for June 6 storm at resolutions Actual L1, Actual L2, Actual L3. \( Q_{\text{shed}} \): total runoff (in.); INFIL: infiltration (in.); E: evaporation (in); ETU: upper zone evapotranspiration (in.); ETD: lower zone evapotranspiration (in.); \( Q_{p,\text{run}} \): peak runoff (cfs); \( Q_{p,\text{out}} \): peak outflow (cfs)………………96

Figure 5.3 Comparison of (a) Infiltration depths, (b) surface runoff depths, (c) peak surface runoffs, and
(d) peak outflows among model resolutions Actual L1, Actual L2 and Actual L3 for the July 27 storm, as a function of spatial variability of different subcatchment parameters……………………………………………………………………98

Figure 5.4 Comparison of ‘F’ and ‘Fs’ at
(a) i = 0.4 in./h and
(b) i = 1.0 in./h for different subcatchments against their respective Ks at different spatial resolutions Actual L1 and Actual L3 considering one time step (=15 min)......................................................................................................................101

Figure 5.5. Peak flows for July 27 storm compared at resolutions Actual L1, Actual L2, Actual L3 under spatial variabilities of different parameters. Ks: saturated hydraulic conductivity; S: slope; dp: depression storage, IMP: percent imperviousness; LOF: length of overland flow...........................................................................................................103

Figure 5.6. Comparison of runoff depths for different subcatchments using spatially variable LOFs for model resolutions Actual L1 and Actual L3 at i = 1.0 in./h considering one time step (=15 min)......................................................................................................................105

Figure 5.7. Comparison of
(a) $F$ and $Fs$ as a function of $K_s$,
(b) actual infiltration depth $f$ as a function of $K_s$, and
(c) surface runoff depth as a function of $LOF$ for subcatchments at resolutions Actual L1 and Actual L3 at rainfall intensity $i = 0.6$ in/h and considering one time step (=15 min)...........................................................................................................108

Figure 5.8. Comparison of
(a) $F$ and $Fs$ as a function of $K_s$,
(b) actual infiltration depth $f$ as a function of $K_s$, and
(c) surface runoff depth as a function of $LOF$ for subcatchments at resolutions Actual L1 and Actual L3 at rainfall intensity $i = 1.0$ in/h and considering one time step (=15 min)...........................................................................................................108

Figure 5.9. Water balance parameters and peak flow rates compared for July 22 storm at resolutions Actual L1, Actual L2, Actual L3. $Q_{shed}$: total runoff (in.); INFIL: infiltration (in.); $E$: evaporation (in.); ETU: upper zone evapotranspiration (in.); ETD: lower zone evapotranspiration (in.); $Q_{p,run}$: peak runoff (cfs); $Q_{p, out}$: peak outflow (cfs)...........................................................................................................109
Figure 5.10. Water balance parameters and peak flow rates compared for July 22 storm resolutions Actual L1, Actual L2, Actual L3. $Q_{\text{shed}}$: total runoff (in.); INFIL: infiltration (in.); E: evaporation (in.); ETU: upper zone evapotranspiration (in.); ETD: lower zone evapotranspiration (in.); $Q_{\text{p,run}}$: peak runoff (cfs); $Q_{\text{p,out}}$: peak outflow (cfs).

Figure 5.11 Effect of increasing length of overland flow on
(a) outflow hydrograph, and
(b) runoff hydrograph for July 22 storm..............................114

Figure 5.12 Effect of increasing length of overland flow on
(a) outflow hydrograph, and
(b) runoff hydrograph for June 2 storm..............................115

Figure 5.13. (a) Original pipes in Actual L1.
(b) A new pipe added and an original pipe length increased in Actual L1........117

Figure 5.14. Effect of increasing conduit length and combined effect of increasing conduit length and length of overland flow on
(a) outflow hydrograph, and
(b) runoff hydrograph for July 22 storm..............................119

Figure 5.15. Effect of increasing conduit length and combined effect of increasing conduit length and length of overland flow on
(a) outflow hydrograph, and
(b) runoff hydrograph for June 2 storm..............................120

Chapter 6. Comparison of Actual and Artificial Networks

Figure 6.1. Percent difference in peak flows predicted by actual and artificial networks at three resolutions L1, L2 and L3 (normalized by the peak flows predicted by the actual networks at each resolution) as a function of total rainfall depth for each storm.................................................123

Figure 6.2. Outflow hydrographs for the storm on April 21 predicted by actual and artificial networks at resolutions
(a) L1,
(b) L2, and
(c) L3.................................................................125
Figure 6.3. (a) Peak runoff and 
   (b) peak outflow compared using Actual L2 and Artificial L2 and combined 
effect of increasing conduit length and length of overland flow for June 2 
storm..............................................127

Figure 6.4. (a) Outflow hydrograph, 
   (b) runoff hydrograph, and 
   (c) water balance parameters for July 27 storm compared using Actual L2 and 
   Artificial L2..............................................129

Figure 6.5. Outflow hydrographs compared at resolutions L1 and L3 using 
   (a) actual networks, and 
   (b) artificial networks for the July 27 storm.........................................131

Figure 6.6. Outflow hydrographs compared at resolutions L1 and L3 using 
   (a) actual networks, and 
   (b) artificial networks for the September 15 storm..................................131

Figure 6.7. Outflow hydrographs compared at resolutions L1 and L3 using 
   (a) actual networks, and 
   (b) artificial networks for the July 22 storm........................................132

Figure 6.8. Outflow hydrographs compared at resolutions L1 and L3 using 
   (a) actual networks, and 
   (b) artificial networks for the June 2 storm........................................133
List of Tables

Chapter 2. Literature Review

Table 2.1 Literature review on the effects of spatial resolution in urban and rural hydrology...............................................................................................................23

Chapter 3. Approach and Detailed Methodology

Table 3.1. Attributes of the output point, line and area shapefiles generated by ANGel, version 1.2........................................................................................................................................50

Table 3.2. List of input raster datasets used for the study area Faneuil Brook sub-basin to calculate SWMM attributes for drainage areas at different resolutions using the Zonal Statistics function in ANGel........................................................................................................69

Table 3.3. List of SWMM input parameter for subcatchments, conduits and junctions that were matched with attributes from the corresponding ANGel output shapefiles.................................................................................................................................72

Chapter 4. Results of Scale Effects

Table 4.1. Differences in hydrologic properties, input parameters and model output between actual and artificial networks at three spatial resolution levels. ........................................77
1.1 Modeling Urban Hydrology

Urban hydrology is defined as the interdisciplinary science of water and its interrelationships with man in an urban watershed (Lazarro 1990). An urban or urbanizing watershed may be defined as one in which impervious surfaces (e.g., roads, parking lots, buildings, etc.) cover or will soon cover a considerable area of the watershed, and where natural flow paths have been substituted by paved gutters, sewers, or other elements of artificial drainage (USDA 1986). In urban areas, due to the intense alteration of natural environmental processes by human activity, the watershed response to precipitation are also significantly altered (e.g. reduced infiltration, decreased travel time, higher runoff, etc.). Urban hydrology involves numerous interacting processes subsystems such as surface runoff, infiltration, conduit routing, groundwater discharge to receiving water, etc. As a result, complex mathematical models are needed to predict or simulate watershed response to rainfall events in urban areas.

Simulation is defined as the mathematical description of the response of a hydrologic system to a series of events for a specific time period (Viessman and Lewis 2003). A hydrologic simulation model, also called a rainfall-runoff model, is a set of equations and algorithms that describe and imitate the behavior of a real system. For example, the model may compute the peak discharge for a real or hypothetical storm. Steps for simulating runoff response for a given watershed usually involve the following: selecting a historic or synthetic rainfall event, specifying a set of input parameters, choosing an appropriate hydrologic/hydraulic model, and determining the output in terms
of an outflow hydrograph and other parameters such as total volume, infiltration, groundwater flow, etc. These models are most commonly used by engineers in design, for example to design a stormwater control device such as a detention pond. They may be used by environmental, conservation, and transportation agencies to compare the hydrologic impact of land development or other watershed changes (e.g., implementation of best-management practices in stormwater management) with existing conditions. They are also used by planners to understand and quantify the impacts of future growth and development related to management decisions.

1.2 Model Subdivision and its Effects

Hydrologic simulation models can be classified as event-based such as the Hydrologic Engineering Center HEC-1 or continuous such as the Stormwater Management Model, SWMM (Rossman 2005). Another classification of urban hydrology models are based on either the lumped parameter approach or the distributed parameter approach (Viessman and Lewis 2003). Whichever model is used, accuracy of model predictions is governed by how well the model mimics the real hydrologic system. Since the urban landscape is characterized by a high degree of heterogeneity, it is important for a model to represent these spatial variations in input parameters. To capture this spatial heterogeneity in input parameters (e.g., land use, slope, soil type, organization of drainage network, etc.), a watershed is divided into smaller sub-basins or subwatersheds, called model subdivision. In such cases, the flow of water through a watershed is modeled by a series of sub-basins that reflect different conditions and routing the runoff hydrographs from each sub-basin outlet through the drainage network to form the runoff hydrograph at the overall basin outlet (Hellweger and Maidment 1999). Some common urban hydrology models that use the distributed-parameter approach are SWMM, InfoWorks, Modeling of Urban Sewers (MOUSE), MIKE URBAN, etc.
Hydrologic models are applied at various spatial resolutions (i.e., sub-basin area) ranging from a hundredth of an acre to hundreds of square miles. After choosing an appropriate hydrologic model, the next task for the modeler is to choose an appropriate level of model subdivision. It has been demonstrated from previous research (Metcalf & Eddy et al. 1971; Wood et al. 1988; Warwick and Litchfield 1993; FitzHugh and Mackay 2000; Ao et al. 2003; Elliott et al. 2009) that spatial resolution does affect model output. A detailed discussion of previous research is provided in Chapter 2. In general, this “scale effect” is the result of non-linearity of model equations. Let us consider a simple illustrative example (Figure 1) to understand how non-linearity and spatial resolution affect model results on an area (two acres) that consists of two equal-sized sub-areas (each one acre). Applying the Rational Method (Figure 1.1a), which is a linear model, on the two areas with runoff coefficients $C = 0.35$ and $0.7$, separately and adding, the resultant peak runoff in response to a 3 in./h storm is 3.15 cfs. Considering the whole area with its average $C = 0.525$, the peak runoff predicted for the same storm is 3.15 cfs. If these peak flows are normalized by their areas, they will lie on a straight line (Figure 1.1c), where the colors red and blue, respectively represent the distributed and the lumped approaches Now, applying the Curve Number (CN) method (Figure 1.1b), which is a non-linear model, on the two areas with $CN = 55$ and $93$ separately and adding, the resultant runoff depth in response to a 2 in. depth storm is 0.66 in. Considering the whole area with its average $CN = 74$, the runoff depth predicted for the same storm is 0.35 in. These results are presented in Figure 1.1d, where the colors red and blue, respectively again represent the distributed and the lumped approaches. The solid markers represent individual runoff depths, while the dashed lines represent average runoff depth for each resolution. From these two examples, it is readily understood that scale does affect results in a non-linear model, and model aggregation predicted a decrease in runoff depths using the CN method.
1.3 Problem Statement

The scaling problem can be easily understood for this simple case. However, modern hydrologic models are considerably more complex. They involve numerous dynamic and interacting non-linear processes and many spatially varying parameters, which when averaged or aggregated are subject to the scale effect. This problem is illustrated from the model results of Zaghloul (1981) presented in Figure 1.2. In this study the dynamic and continuous Stormwater Management Model (SWMM) was applied on an urban catchment in Sunshine City, Australia. Two levels of spatial discretization using 12 and 5 subcatchments were used, and the predicted peak flows at these resolutions were significantly different. This observation leads to different
questions that the modeler would need to address such as why the peak flows are
different at different resolutions, will the model predictions at different resolutions show a
different trend for another storm, and which model resolution will best predict the
observed data.

In the urban environment hydrologic processes operate over a range of spatial
scales. Surface runoff from a pavement travels a few meters or less before it enters a
catchbasin, and pipe networks transfer water over kilometers or more. One solution to
the scaling problem is to consistently simulate at a very high resolution. However, given
current computing power and data availability, modeling every pipe, gutter, driveway,
etc. is feasible only for small areas. Therefore, it is important for the modelers to
understand how different levels of spatial aggregation affect model results, and how
these effects may be different for different models.

Figure 1.2. Flow hydrographs predicted by applying SWMM to an urban catchment at two different spatial
resolutions in a study conducted by Zaghloul (1981).
1.4 Research Objectives and Approach

The main objective of this study is to characterize and understand the effects of spatial resolution on model results for an urban catchment. A significant number of projects have evaluated scale effects in rural hydrology. However, limited investigations have been conducted in urban areas. Among these studies, a general consensus on how spatial resolution affects model results has not been reached. Also, the underlying mechanism(s) behind such effects were not typically explored. This study is an attempt to bridge that gap in terms of understanding the scale effect, and to investigate the mechanism(s) that produce consistent or different effects in model output. Compared to the earlier studies that evaluated scale effects with respect to one or two storm events, the aim of the present research is to assess the impacts of spatial resolution with respect to differences in storm characteristics. It is important to differentiate scale effects among different storms because while larger storms may be used for design of flood-control structures, the smaller events may be more important in terms of water quality effects.

The general approach adopted in this study is to develop models for an urban catchment at different levels of aggregation, perform hydrologic simulations using these resolutions, and compare model results with respect to annual total flow volume and peak flows for different storms. A program is developed to generate artificial sewer networks (based on a fractal algorithm) called Artificial Network Generator (ANGel). The motivation for developing this program is to analyze existing drainage networks and generate artificial networks at different resolutions, along with the hydrologic/hydraulic attributes of corresponding drainage areas and nodes. This will enable a faster spatial scaling analysis compared to the manual aggregation/disaggregation techniques. The present study builds on an existing model of the Faneuil Brook subbasin in the lower Charles River watershed in Boston set up by Zarriello and Barlow (2002). Different
versions of this model with various spatial resolutions were developed, varying from relatively coarse (e.g. 4 subcatchments) to very dense (e.g. 401 subcatchments). The hydrologic simulations were performed using the EPA Storm Water Management Model, SWMM (Rossman 2005).

1.5 Thesis Organization

This section briefly describes the structure of the thesis. An extensive literature review of scale effects in both rural and urban hydrology is presented in Chapter 2. These results are summarized, a “missing link” in previous research is identified, and the open questions that are addressed by this research are stated.

Chapter 3 (titled “Approach and Detailed Methodology”) first discusses the general approach that has been adopted in this study for analysis of spatial resolution effects. This chapter describes the development of models at different spatial resolutions using actual sewer networks and networks generated artificially based on a fractal algorithm. The study area used was the Faneuil Brook sub-basin, which is a subwatershed of the lower Charles River watershed. The chapter discusses in detail the program Artificial Network Generator (ANGel) that was developed as part of this research, the underlying methodology based on fractal geometry, presents all the input options available to the user in the form interface to ANGel, and also explains the program structure behind ANGel. It also provides the definitions of the various SWMM input parameters that are generated using ANGel, and discusses the approach used to estimate these input parameters for the sub-basin area used in this study. Finally, it describes how ANGel output is imported into the Stormwater Management Model (SWMM).
Chapter 4 (titled ‘Results of Scale Effects”) discusses the scale effects observed with respect to annual total outflow volume and peak flows for different storm events in water years 1999, 2000 and 2001 using both actual and artificial networks. This chapter is divided into broad sections: results of scale effects in large storms and results of scale effects in small storms.

Chapter 5 (titled “Analysis of Scale Effects”) provides a detailed analysis of the model results provided in Chapter 4. This chapter identifies the parameters and discusses the underlying mechanism(s) that are responsible for scale effects in large and small storms.

Chapter 6 (titled “Comparison of Actual and Artificial Networks”) presents a comparison of peak flows and scale effects using actual and artificial networks at different spatial resolutions for different storm events. It also provides an analysis of scenarios when either the peak flows or the scale effect using artificial networks were not consistent with those using actual networks.

Chapter 7 summarizes and presents the main conclusions of this research. Finally, some further studies related to the spatial scaling analysis are also recommended here.
1.6 References


Chapter 2. Literature Review

It has already been established in Chapter 1, using simple illustrative examples, that spatial resolution does affect results of hydrologic simulations. The effect of spatial scale has been recognized in the hydrologic community since the early 1960s (Minshall 1960; Amorocho 1961). The general approach used to investigate the scale effect was to model processes at different spatial resolutions and analyze the hydrologic response predicted by these model resolutions. Model results were compared in terms of water quantity parameters such as peak flow, total runoff volume, and water quality parameters such as peak sediment load, total nitrogen and total phosphorous. This chapter is divided into three sections. The first section presents a brief overview of previous research on scale effects in rural hydrology. The second section discusses the effects of performing model simulations at varying spatial resolutions in urban hydrology. The third section summarizes the results from previous research and lists the open questions that are addressed by this research.

2.1 Rural Hydrology

Significant research effort has focused on investigating the effect of spatial scale and resolution in rural hydrology (e.g., Freeze 1980; Dooge 1981; Sivapalan et al. 1987; Wood et al. 1988; Song and James 1992; Gupta et al. 1994; Wolock and Price 1994; Zhang and Montgomery 1994). In one of the earlier studies conducted on the scaling problem, Sivapalan et al. (1987) studied the effect of spatial variability of rainfall and soil properties on a physically based conceptual runoff model. They identified five dimensionless catchment similarity parameters (e.g. scaled saturated hydraulic
conductivity) and concluded that any two subcatchments will produce similar hydrologic response if they are identical in these five parameters, irrespective of the spatial variability of individual input parameters among them. In most of the studies conducted on the effects of spatial resolution on model results, the approach used was to vary either the number of watershed subdivisions or the input data resolution and observe the effect on model results. This review of previous research on spatial resolution effects in rural hydrology will be discussed based on these two approaches.

Goodrich et al. (1988) studied the effects of watershed subdivision on a semiarid experimental watershed by using four levels of model aggregation with 1, 9, 25, and 30 elements. They applied the KINEROS model and observed the results in terms of runoff volumes and peak flows for three different storm events, which were classified as small, medium and large based on their return period. For the largest storm (~2.8 y return period, small flood), there was very little difference in runoff volumes and peak flows between the different aggregation levels since this storm was large enough to overwhelm the variability of infiltration and geometric properties represented in the most detailed system. For the small and medium storms, both the runoff volumes and peak flows predicted decreased (maximum 18% and 10%, respectively) with aggregation from the highest to the lowest resolution. Wood et al. (1988) applied a modified version of TOPMODEL to the Coweeta River catchment subdivided into 3, 19, 39, and 87 sub-basins and their results indicate that model output in terms of cumulative infiltration volume, runoff rate, etc. was very different between above and below a mean subcatchment area of 1.0 km². They observed that the variability of subcatchment responses with increasing subcatchment area decreased considerably and almost stabilized above a mean subcatchment area of 1.0 km². They identified this threshold as a Representative Elementary Area (REA), which they concluded was strongly influenced
by topography and variabilities of soils and rainfall inputs. Song and James (1992) defined the optimal scale as “the size of a unit or subcatchment within which the hydrologic response to rainfall can be treated as homogeneous and still have runoff simulation reproduce the hydrologic response of the catchment”. To determine this scale, they applied a version of the Stanford Watershed Model to catchments at various resolutions (e.g. 1, 4, 9, 16, 25, 36 subdivisions) to the Bodfish Creek catchment. They found that minimum values for one of their objective functions occurred for 25 subdivisions, or a mean sub-basin area of 0.30 mi². Considering an additional objective function and two other catchments lead them to conclude that the “optimal scale” is between 0.3 to 2.1 mi² (0.8 to 5.4 km²).

Norris and Haan (1993) used the HEC-1/SCS model to evaluate the effect of watershed subdivisions (1, 2, 5, 10, 15 subdivisions) on the Little Washita watershed near Chicasa in Oklahoma. They evaluated the effects for one 24-h 50-y synthetic storm and found that peak flow decreased with aggregation, and this rate of decrease of peak flow increased below 5 sub-basins. However, runoff volumes varied by only 2.5% over the entire range of subdivisions. Bingner et al. (1997) used seven cases of watershed subdivisions (from 14 to 470 subdivisions) on the Goodwin Creek watershed in northern Mississippi. They applied the SWAT model and concluded that annual total runoff volumes were insensitive to watershed subdivision. On the other hand, fine sediment yield decreased with aggregation, which was directly co-related to decrease of both slope and crop land. A similar study was conducted by FitzHugh and Mackay (2000) on the agricultural Pheasant Branch watershed in Wisconsin using the SWAT model. They used eight watershed delineations (from 3 to 181 subdivisions) and found that annual, monthly and daily streamflows decreased at coarser watershed delineations. Annual,
monthly and daily sediment yield at outlet showed slight increase with aggregation, while the total sediment generated increased significantly with aggregation.

A similar conclusion as the Song and James (1992) study was reached by Ao et al. (2003). They applied their hydrologic model BTOPMC (Block-wise use TOPMODEL with Muskingum-Cunge routing) to the Fuji-Kawa and Nakagawa basins in Japan using various spatial resolutions ranging from 1-249 subdivisions for Fuji-Kawa and 1-153 subdivisions for Nakagawa basins. Their results showed that the peak flows and baseflows during flood period decreased as spatial resolution decreased. On the other hand, during dry season baseflow increased as spatial resolution decreased. Due to this, the annual runoff was insensitive to spatial resolution. They proposed a threshold subbasin scale beyond which the differences in model results at different spatial resolutions are quite low. This threshold level was a function of sub-basin and structure of stream networks, and it was different for the basins studied.

The next four studies conducted in the recent past, which were reviewed, all applied the SWAT model to different rural watersheds by varying the number subdivisions. Jha et al. (2004) studied four agricultural watersheds in Iowa and concluded that average annual streamflow was insensitive to watershed subdivision. However, annual sediment load, nitrate and mineral phosphorous concentrations were not sensitive to aggregation up to 17 subwatersheds, and then decreased with aggregation. They proposed a threshold area for each output parameter studied, beyond which increasing the subdivision had very little impact on model results. Arabi et al. (2006) studied scale effects, with and without using BMPs on two watersheds in the Maumee River basin in Indiana with subdivisions varying from 1-89 and 1-103. Without BMPs sediment yield at outlet, total phosphorous and total nitrogen decreased significantly with aggregation. With BMPs while sediment yields was insensitive to
watershed subdivision, total phosphorous and total nitrogen yields increased with aggregation. Tripathi et al. (2006) studied the effects of watershed subdivision on the agricultural Nagwan watershed in eastern India by dividing it into 1, 12, 22 sub-basins. Using SWAT they evaluated the impacts of subdivision on annual total runoff volumes, as well as, on water balance components like evapotranspiration, percolation and soil water content. They concluded that annual total runoff volumes were insensitive to watershed subdivision. However, monthly runoff peaks were significantly lower at lower resolution. Also, evapotranspiration decreased, while percolation increased with aggregation. However, these differences were the most significant for aggregation from 22 sub-basins to 1 or from 12 sub-basins to 1. Muleta et al. (2007) used 6 levels of subdivision (9-118 sub-basins) on the Big Creek watershed in Southern Illinois. They concluded that flow generated in the form of surface runoff, lateral flow, groundwater flow and streamflow exiting the watershed decreased with spatial aggregation. Also, the sediment generated and sediment exiting the watershed decreased significantly at coarser resolutions. However, the percentage of generated sediment that was deposited in the reaches increased as the spatial resolution decreased. A recent study was conducted by Cleveland (2009), which applied the HEC-HMS program to five watersheds in central Texas using different subdivisions (1, 2, 3, 5, 7). They found that spatial resolution did not significantly affect the peak discharge or the total runoff volumes predicted at these resolutions.

Among previous research in rural hydrology that studied scale effects on model results by varying resolution of the input data, one of the earliest studies was conducted by Wolock and Price (1994). They used the TOPMODEL and studied three different spatial resolutions by varying the DEM map scale and data resolution. They found that at the coarsest resolution (90 m DEM at 1:250,000), the daily peak flow and the ratio of
overland flow to total flow were the highest. However, the mean depth to the water table was the lowest for the coarsest resolution. A similar study was conducted by Zhang and Montgomery (1994), on two catchments in Oregon and California using TOPMODEL. They compared five DEM resolutions (2, 4, 10, 30, 90 m) and observed the results with respect to peak flows for two storms. For the smaller storm, for both catchments, the coarsest resolution predicted a higher peak flow. For the larger storm, in the case of one catchment, the peaks predicted by the different DEMs were very similar. For the same storm, in case of the other catchment, the coarsest DEM resolution predicted the highest peak flow, while that predicted by the finest DEM resolution was in between.

Kalin et al. (2003) studied the effects of spatial resolution on peak flows, peak sediment discharges and total sediment loads of big rainfall events, for two experimental field scale USDA watersheds near Treynor, Iowa by applying the KINEROS model. They varied the spatial resolution by varying the geomorphologic resolution of the drainage network and concluded that peak runoff decreased with aggregation. However, beyond an optimal resolution, the differences in peak runoff rates become insignificant. Peak sediment discharge and total load of sediment at first increased or remained almost constant with aggregation, and after a threshold resolution is reached, they decreased steadily with aggregation.

The next two studies both varied scale by varying input DEM data resolution and used the SWAT model. Chaplot (2005) varied the spatial resolution of the lower Walnut Creek watershed in central Iowa by varying the input DEM resolution from 20 m to 500 m, and also by varying the soil data resolution. They concluded that annual runoff volumes had low impact by varying DEM resolution, but DEMs over 90 m were poor in predicting SWAT runoff estimates. Sediment and nitrate loads also decreased at coarser resolution. The study conducted by (Chaubey et al. 2005) used seven input DEM
resolutions and compared results at these resolutions with respect to annual flow volume, nitrate and total phosphorous loads for an agricultural watershed in northwest Arkansas. They found that stream flow volume and nitrate load decreased at coarser DEM resolutions. However, total phosphorous increased when 100, 150 and 200m resolutions were used, and decreased for other DEM resolutions. A similar finding was made by Di Luzio et al. (2005), who varied both the number of watershed subdivisions and also the input DEM resolution. They observed that yearly runoff volumes and yearly sediment yields decreased at lower subdivisions and coarser DEM resolution.

The next two studies followed a similar approach to study the effects of spatial resolution by varying the input soil data. Peschel et al. (2006) used two cases, each with 29 sub-basins for a watershed in the upper Sabinal River in Texas. In one case they used the coarser (1/250,000) STATSGO data with 491 Hydrologic Response Units (HRUs), and in the other case they used the finer resolution (1/25,000) SSURGO data with 2397 HRUs. They observed that daily mean surface runoff volume and daily mean evaporation were higher at lower resolution; while daily mean percolation and daily mean groundwater return flow were lower at the lower resolution. They concluded that significantly higher values of saturated hydraulic conductivities associated with SSURGO soils caused the SSURGO model to predict higher water yields. Geza and McCray (2008) applied a similar approach to the mountainous arid Turkey Creek watershed in Denver. They used two cases each with 61 sub-basins. In one the coarser STATSGO data with 261 HRUs were used, and in the other finer resolution SSURGO data with 1301 HRUs were used. They found that monthly average streamflow and annual average dissolved nitrogen loading were lower at the coarser resolution, while annual average sediment loading and annual average organic nitrogen loading were higher at the coarser resolution.
There were some studies that varied both the number of watershed subdivisions and the input data resolution. For example, Di Luzio et al. (2005) conducted a study on the agricultural Goodwin Creek watershed in the Yazoo River basin of Mississippi, where they used different combinations of input data resolutions such as DEM, land cover and soil map. This broadly resulted in two cases: a coarser resolution with 90 m DEM and 200 subdivisions, and a finer resolution with 30 m DEM and 248 subdivisions. They observed that yearly runoff volumes and yearly sediment yields decreased at the lower resolution. Kumar and Merwade (2009) studied the effects of spatial resolution on SWAT model output on the primarily rural St. Joseph River watershed at the intersection of IN, MI and OH. They varied both the number of sub-basins and the scale of the input soil data by using the coarser STATSGO data and the finer SSURGO data. They found that daily average streamflow could increase or decrease at lower resolutions. However, monthly and annual average streamflows were not sensitive to spatial resolution.

2.2 Urban Hydrology

Compared to the vast literature available on the effects of spatial resolution in rural hydrology, limited investigations have focused on the scale effect in urban areas. One of the earliest studies that evaluated the effects spatial resolutions on model results in urban hydrology was the report by Metcalfe and Eddy (1971), which applied the Stormwater Management Model (SWMM) to the Northwood area in Baltimore. Since detailed information for the subdivision of the drainage basin was not available in this area, the feasibility of utilizing a coarser subdivision of the drainage system was investigated. It was observed in this study that aggregating a 12-subcatchment system to a 5-subcatchment system resulted in a lower peak flow prediction, which was also
Another early study in urban hydrology was conducted by Proctor and Redfern (1976). They applied SWMM to four hypothetical test areas. In each they used three levels of model subdivision with 37, 5 and 1 subcatchments. Five synthetic storms were applied to each of these resolutions for each area and peak flows and total storm runoff volumes were compared. They observed no differences among the model resolutions for any of the areas with respect to the total runoff volume. This was consistent for all five storms. However, the results were different for peak flows. For the intense one-hour storm, peak flows predicted increased with model aggregation. For the two-hour storm, again the coarsest resolution model predicted the highest peak flow, but the intermediate resolution predicted the lowest peak flow. For the other three storms of longer duration and lower intensity, there were no significant differences in peak flows among the different resolutions for any of the four areas.

Zaghloul (1981) investigated the effects of spatial resolutions on a hypothetical urban area and on four real test areas in Chicago, Baltimore, Australia and Canada. The hypothetical test area was 634 acres (2.6 sq. km) and was discretized into 37, 5 and 1 subcatchments, and a 2-hour synthetic storm with a peak intensity of 2 in./h was used for the simulation. For this area, Zaghloul (1981) observed a slight increase in peak flow with model aggregation. The 10-acre (0.04 sq. km) urban area in Chicago was discretized into 80 subcatchments and then lumped as one area. For this small area, the hydrograph predicted by the lumped system was slightly higher than the detailed system. The Vine street catchment in Australia was 173 aces (0.7 sq. km). This area was discretized into 51 and 12 subcatchments and a real storm event was used for simulation. The results showed that model aggregation for this area significantly increased the peak flow by as much as 33%. The total area of the Banatyne catchment in Canada was 542 acres (2.2 sq. km) and this area was discretized into 41, 3 and 1
subcatchments. A real storm event was used for simulation and results showed that peak flow increased due to aggregation and was by 20% higher for the lumped system than the detailed system. Therefore, Zaghloul (1981) concluded that in four out of the five test areas, similar results were observed and peak flows increased with model aggregation. He proposed that the increase in peak flow with lower subcatchments was due to lower loss in conduit storage, which could be accounted for by reducing the hydraulic width for increased surface storage in case of lower subdivisions. However, he observed the opposite effect in the Northwood area of Baltimore. This urban area of 47.4 acres (0.2 sq. km) was discretized into 12 and 5 subcatchments (consistent with Metcalf and Eddy 1971). For this area, Zaghloul (1981) observed that peak flow decreased with model aggregation (consistent with Metcalf and Eddy 1971). However, the author did not explain the reason behind this observed discrepancy in peak flows with model aggregation between this area and the earlier four areas.

Stephenson (1989) conducted a study on the urban catchment in Sunninghill, north of Johannesburg, South Africa. He used the WITWAT model to investigate spatial scale effects on runoff volume and peak flow rate at four resolutions, coarse, medium, fine and very fine. He observed that peak flow rate increased and predicted runoff volume decreased with model aggregation. He also observed that while the finest resolution model predicted runoff volume more accurately than coarser models, the coarser models predicted peak flow rates better. This study discussed how infiltration rate could be adjusted in the coarsest level to match runoff volume with measured data, and how the overland flow path length or Manning's roughness coefficient could be adjusted for better prediction of peak flow rates by the finer levels of discretization. Warwick and Litchfield (1993) used the EPA SWMM to study the effects of subdividing an urban residential area in Reno, Nevada. They evaluated the effect by dividing the
area into 30, 4 and 1 sub-basins and compared the peak flow for one storm at these resolutions. They found that the peak predicted by the lowest resolution model was lower than the peak flow predicted by the highest resolution, but higher than the peak flow predicted by the model at intermediate resolution. This means that peak flow first decreased, and then later increased with model aggregation. However, the time to peak was the worst for the lowest resolution.

Kronaveter et al. (2001) applied the model HMM on a synthetic urban neighborhood, similar to a residential neighborhood in Israel’s coastal plain. He used three spatial resolutions: macro, mezzo and micro and observed that for soils with high saturated hydraulic conductivity (>30 mm/h), the differences in infiltration among the three resolutions were small (less than 2.5% of annual rainfall). However, for soils with lower saturated hydraulic conductivity (<30 mm/h), the predicted infiltration increased with spatial aggregation.

Among the recent studies that evaluated the effects of spatial resolution on urban hydrology, Park et al. (2008) applied SWMM to analyze the effects of aggregation on total discharge volume and pollutant loading in terms of suspended solids and BOD. They conducted their study in an experimental urban area by dividing it 39, 38, 23 and 1 sub-basins and analyzed two storm events. The different resolutions produced almost identical hydrographs for both the storms and they concluded that the total discharge volume was practically insensitive to spatial resolution. However, there was a significant decrease in accumulated BOD load and in the suspended solids load from runoff as a result of model aggregation.

Another interesting study was conducted by Elliot et al. (2009), who analyzed the effects of aggregation of on-site stormwater control devices in an urban catchment. The 0.83 km² study area was located in the Tangatu-Stream catchment in Waitekere city,
Auckland, New Zealand and the dynamic model for urban stormwater improvement conceptualization (MUSIC) was used. The detailed model set-up was divided into 810 source areas, each with its hypothetical stormwater device, which were aggregated to 55 sources, 7 sources and a single source, along with their corresponding devices and drainage network. The aggregation technique was applied when no devices, and when detention, infiltration and bioretention devices were used. The results were analyzed for a 1-h 25 mm storm event, and the effect of aggregation was studied with respect to parameters such as peak flow, base flow, mean flow, and maximum concentration of total suspended solids. They found that aggregation had negligible effect on mean flow, base flow, maximum concentration and total contaminant load of suspended solids using none or any of the stormwater control devices. However, aggregation to a single source resulted in an increase of peak flow by 34%, 38% and 24%, respectively when no devices, bioretention devices and infiltration devices were used. They suggested that the flow attenuation occurred in the higher resolution models due to aggregation of the drainage network, which results in variability of travel times within the network.

2.3 Summary and Open Questions

From the above review of previous research on the effects of spatial resolution on model results, it is concluded that significantly more projects focused on rural hydrology compared to urban areas. It is also well-established that scale (level of subdivision of model elements, input data resolution, or both) does affect both water quantity and water quality predictions. Since the present research focuses only on water quantity parameters, specifically total outflow volume and peak flow, the results of
previous research will be summarized in this section with respect to only these parameters.

It is clear from previous studies that a general consensus about how the level of spatial aggregation influences model predictions is not quite evident. While some studies conclude that runoff volume was insensitive to scale (Ao et al. 2003; Chaplot 2005; Tripathi et al. 2006; Park et al. 2008), other studies observed that runoff volume decreased due to model aggregation (Stephenson 1989). A similar discrepancy was observed with respect to peak flows. Kalin et al. (2003) and Tripathi et al. (2006) found that peak flow decreased with model aggregation, while Stephenson (1989) and Elliot et al. (2009) observed significantly higher peak flows as a result of model aggregation. On the other hand, there were some studies that observed both of these effects with respect to peak flows in different scenarios (e.g., Zaghloul 1981; Wood et al. 1988; Warwick and Litchfield 1993; Zhang and Montgomery 1994; Ao et al. 2004). Wood et al. (1988), Zhang and Montgomery (1994) and Ao et al. (2004) also identified a threshold level of model subdivision or input data resolution, beyond which model results tend to stabilize. Zaghloul (1981) observed that peak flow decreased with increasing resolution for four watersheds, and increased with increasing resolution for one study area. A shorter version of the literature review is included in Table 2.1.
<table>
<thead>
<tr>
<th>U/R</th>
<th># Areas</th>
<th># subdivisions</th>
<th>Model</th>
<th>Output and Scale Effect*</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>1</td>
<td>4, 19, 401; 5, 21, 616</td>
<td>SWMM</td>
<td>$Q_{\text{tot}} \leftrightarrow Q_p$ ↑</td>
<td>This study</td>
</tr>
<tr>
<td>U</td>
<td>1</td>
<td>1, 7, 55, 810</td>
<td>MUSIC</td>
<td>$Q_{\text{avg}} \leftrightarrow Q_{\text{base}} \leftrightarrow Q_p$ ↑</td>
<td>Elliot et al. (2009)</td>
</tr>
<tr>
<td>U</td>
<td>1</td>
<td>1, 23, 38, 39</td>
<td>SWMM</td>
<td>$Q_{\text{tot}} \leftrightarrow$</td>
<td>Park et al. (2008)</td>
</tr>
<tr>
<td>U</td>
<td>1</td>
<td>1, 6, 47</td>
<td>HMM</td>
<td>INFILannual ↑</td>
<td>Kronaveter (2001)</td>
</tr>
<tr>
<td>U</td>
<td>1</td>
<td>1, 4, 30</td>
<td>SWMM</td>
<td>$Q_p$ ↓ optimal resolution</td>
<td>Warwick and Litchfield (1993)</td>
</tr>
<tr>
<td>U</td>
<td>1</td>
<td>4 levels</td>
<td>WITWAT</td>
<td>$Q_{\text{tot}}$ ↓</td>
<td>Stephenson (1989)</td>
</tr>
<tr>
<td>R</td>
<td>5</td>
<td>1, 5, 37; 1, 80; 5, 12; 12, 51; 1, 3, 41</td>
<td>SWMM</td>
<td>$Q_p$ ↑, $Q_p$ ↔</td>
<td>Zaghloul (1981)</td>
</tr>
<tr>
<td>U</td>
<td>1</td>
<td>1, 2, 3, 5, 7</td>
<td>HEC-HMS</td>
<td>$Q_p$ ↔, $Q_{\text{tot}}$ ↔</td>
<td>Cleveland et al (2009)</td>
</tr>
<tr>
<td>R</td>
<td>2</td>
<td>10-97; 7-41</td>
<td>SWAT</td>
<td>$Q_{\text{avg,monthly}}$ ↓</td>
<td>Kumar and Merwade (2009)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>2 resolutions</td>
<td>SWAT</td>
<td>$Q_{\text{avg,monthly}}$ ↓</td>
<td>Geza and McCray (2008)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>9-118 (6 cases)</td>
<td>SWAT</td>
<td>$Q_{\text{shed,annual}}$ ↓, $Q_{\text{avg,annual}}$ ↓</td>
<td>Muleta et al (2007)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>2 resolutions</td>
<td>SWAT</td>
<td>$Q_{\text{tot,annual}}$ ↑, $Q_{\text{avg, monthly}}$ ↓, $Q_{\text{avg, annual}}$ ↔</td>
<td>Peschel et al. (2006)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>1, 12, 22</td>
<td>SWAT</td>
<td>$Q_{\text{tot,annual}}$ ↔, $Q_p$ monthly ↓, $ET_{\text{annual}}$ ↓, $PERC_{\text{annual}}$ ↑</td>
<td>Tripathi et al. (2006)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>12 DEM resolutions</td>
<td>SWAT</td>
<td>$Q_{\text{tot,annual}}$ ↔</td>
<td>Chaplot (2005)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>7 DEM resolutions</td>
<td>SWAT</td>
<td>$Q_{\text{tot,annual}}$ ↓</td>
<td>Chaubey et al. (2005)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>12 resolutions</td>
<td>SWAT</td>
<td>$Q_{\text{tot,annual}}$ ↓</td>
<td>Di Luzio et al. (2005)</td>
</tr>
<tr>
<td>R</td>
<td>2</td>
<td>1-249; 1-153</td>
<td>BTOPMC</td>
<td>$Q_p$ ↓ threshold subdivision, $Q_{\text{tot,annual}}$ ↔</td>
<td>Ao et al. (2004)</td>
</tr>
<tr>
<td>R</td>
<td>4</td>
<td>5-53; 3-47; 3-35</td>
<td>SWAT</td>
<td>$Q_{\text{avg,annual}}$ ↔</td>
<td>Jha et al. (2004)</td>
</tr>
<tr>
<td>R</td>
<td>2</td>
<td>4 levels for each</td>
<td>KINEROS</td>
<td>$Q_p$ ↓, $Q_p$ ↔ after optimal resolution.</td>
<td>Kalin et al. (2003)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>3-181 (8 cases)</td>
<td>SWAT</td>
<td>$Q_{\text{avg,annual}}$ ↓, $Q_{\text{avg,monthly}}$ ↓</td>
<td>FitzHugh and Mackay (2000)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>14 – 470 (7 cases)</td>
<td>SWAT</td>
<td>$Q_{\text{tot,annual}}$ ↔</td>
<td>Bingner et al. (1997)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>3 resolutions</td>
<td>TOPMODEL</td>
<td>$Q_p$ daily ↑</td>
<td>Wolock and Price (1994)</td>
</tr>
<tr>
<td>R</td>
<td>2</td>
<td>5 DEM resolutions</td>
<td>TOPMODEL</td>
<td>$Q_p$ ↓ optimum DEM resolution</td>
<td>Zhang and Montgomery (1994)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>1, 2, 5, 10, 15</td>
<td>HEC-1/SCS</td>
<td>$Q_p$ ↓, $Q_{\text{tot}}$ ↔</td>
<td>Norris and Haan (1993)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>4 levels</td>
<td>KINEROS</td>
<td>$Q_p$ ↔, $Q_p$ ↓, $Q_{\text{tot}}$ ↓</td>
<td>Goodrich (1988)</td>
</tr>
<tr>
<td>R</td>
<td>1</td>
<td>3, 19, 39, 87</td>
<td>TOPMODEL</td>
<td>$Q_{\text{tot}}$ ↓, $Q_p$ ↑, INFIL ↑</td>
<td>Wood et al. (1988)</td>
</tr>
</tbody>
</table>

U = urban; R = rural; $Q_p$ = peak flow; $Q_{\text{tot}}$ = total flow volume; $Q_{\text{base}}$ = baseflow; $Q_{\text{shed}}$ = surface runoff; $Q_{\text{avg}}$ = average streamflow; INFIL = infiltration volume; ET = evapotranspiration; PERC = percolation; GWFLW = groundwater flow; LF = lateral flow. Unless specified annual, monthly or daily, all above parameters refer to individual storms.

* as resolution decreases; ↓ = decreases; ↑ = increases; ↔ dual effect ; ↔ = negligible or no effect
Although the effects of spatial resolution have been documented in urban hydrology, the underlying mechanism(s) were not typically explored. Also, the studies that observed a dual scale effect in peak flows did not explain this discrepancy. Some studies in rural areas (e.g., Zhang and Montgomery 2004) compared model results for two different storm events, a large storm and a smaller storm. However, studies in urban areas did not analyze model results with respect to differences in storm characteristics. Finally, some studies (e.g. Peschel et al. 2006; Tripathi et al. 2006; Muleta et al. 2007) analyzed the effects of spatial resolution on total runoff volume with respect to other water balance components such as has evapotranspiration, infiltration and groundwater flow. A similar approach was not adopted for the urban areas.

This summary of previous work leads to some key open questions that are addressed by this research. The first question is to determine how spatial resolution affects results predicted using a dynamic rainfall-runoff model (e.g. SWMM) on an urban catchment. This implies examining how model results such as total outflow volume and peak flows compare at different levels of spatial aggregation. Since it was observed from the literature that this scale effect can go in both directions (i.e., peak flows may either increase or decrease with spatial aggregation), the next question is to determine if the scale effects are different for different storm characteristics. Finally, the most important question that arises from the review of previous research is about the mechanism(s) of the scale effect. This leads to determining how impact of spatial resolution on model results affect other water balance components such as infiltration, evapotranspiration, etc. Overall, these open questions define the main objective of this study, which is to understand the principal mechanism(s) leading to the scale effect in an urban hydrologic model.
2.4 References


Chapter 3. Approach and Detailed Methodology

The general approach is to develop models at various spatial resolutions, perform simulations and observe differences in predicted total flow volume and peak flow. We use both actual and artificial (generated using a fractal algorithm) networks. The study builds on an existing model of the Faneuil Brook subbasin in the lower Charles River watershed in Boston set up by Zarriello and Barlow (2002). The original model was divided into eighteen subcatchments. Different versions of this model with various spatial resolutions were developed, varying from relatively coarse with four subcatchments to the densest with 401 subcatchments. Artificial models with different spatial resolutions were developed using the program Artificial Network Generator, ANGel (Ghosh et al. 2006). This program and its various features will be described in detail in subsequent sections. The artificial models varied from relatively coarse with five subcatchments to the densest with 616 subcatchments. Corresponding hydrologic models were developed for each of these networks and model output was compared for 90 different storm events (in water years 1999, 2000 and 2001) in terms of peak flow and total flow volume. The hydrologic simulations were performed using the EPA Storm Water Management Model, SWMM (Rossman 2005). SWMM is a dynamic rainfall-runoff simulation model used for single event or long-term (continuous) simulation of runoff quantity and quality in storm, sanitary and combined sewers from primarily urban areas. The original model set up by Zarriello and Barlow (2002) used PCSWMM 2000 (James et al. 2005), a proprietary interface to SWMM developed by Computational Hydraulics International (CHI) with the computational engine version SWMM 4.4. The interface used in the present study is PCSWMM 2010 with the engine SWMM 5.0.016.
3.1 Study Area

The study area used in this research is the Faneuil Brook sub-basin. Faneuil Brook is a tributary to the lower Charles River in Boston. The lower Charles River extends approximately 15 km (Zarriello and Barlow, 2002) between the Watertown Dam and the New Charles River Dam in Boston, Massachusetts. The four largest tributaries to the lower Charles River, excluding the watershed above Watertown Dam (upper watershed), are Laundry Brook, Faneuil Brook, Muddy River, and Stony Brook. The lower Charles River Watershed is one of the oldest urban areas in the United States, and the Faneuil Brook sub-basin covers parts of the municipalities of Newton and Boston. The drainage area of the Faneuil Brook sub-basin is approximately 4.66 km². Figure 3.1 shows the Faneuil Brook sub-basin including its sub-catchments, the actual sewer lines and Chandler Pond.

Figure 3.1. Faneuil Brook sub-basin of the lower Charles River watershed, showing the actual sewer network and the pond (Zarriello and Barlow, 2002).
The Faneuil Brook sub-basin is a part of the greater Boston area and is highly urbanized. The land use in this area is represented primarily by multi-family residential (49 percent) and single-family residential (16 percent) uses. Other land uses include participation recreation, such as golf and tennis courts, playgrounds (10 percent), urban open space, such as athletic fields, cemeteries and parks (9 percent), commercial (9 percent), forest (3 percent), transportation (2 percent), open water (1 percent), and industrial (1 percent). The major soil types in the Faneuil Brook sub-basin are either derived from till or are disturbed urban land. Soils classified as disturbed urban are found mostly near the river in areas filled to eliminate tidal marshes and mudflats. Another soil type found to the north of the sub-basin is derived from glacial outwash (Zarriello and Barlow, 2002). The two geomorphic districts in the Charles River watershed are the Boston Lowland and the Needham Upland. The Faneuil Brook sub-basin, which lies in the lower Charles River watershed, is in the Boston Lowland. The Boston Lowland is less than 50 ft above sea level and historically comprised large areas of mudflats and tidal marshes (Zarriello and Barlow, 2002).

### 3.2 Input Data Preparation and Modifications to Original Model

The existing “actual” sewer network of the Faneuil Brook sub-basin was obtained from the USGS GIS database of the lower Charles River (Zarriello and Barlow, 2002). First, it was ensured that all the lines in the shapefile pointed in the upstream direction. Note that the most downstream point of the Faneuil Brook sub-basin is located at the outfall point to the lower Charles River in the northern part of the sub-basin. A number of modifications were made to the existing network and these are discussed next.

All loops in the existing actual networks were opened up. Double sewer lines were removed from the existing network. The double sewer lines run parallel to each
other being on both sides of the Massachusetts Turnpike, which runs along the northern part of the Faneuil Brook sub-basin. The present version of ANGel is not capable of handling such double sewer lines, and hence one of the double lines was removed.

In the original model, different precipitation gages were used for different subbasins. However, to remove the effects of spatially variable rainfall, all subcatchments were assigned to the same gage here. The original model included a pond as a storage element, which was replaced by a subcatchment of equal area. This simplification was necessary since the program developed to generate artificial networks does not support storage elements. Removing the pond caused the peak flow to be attenuated, as will be discussed later. However, since this occurs at all spatial resolutions, this modification does not influence the results of the scaling analysis. The saturated hydraulic conductivity data used in the original model was obtained from the 1995 SSURGO database, the format of which was based on the older State Soil Survey Database (SSSD). This version was later modified based on the National Soil Information System (NASIS) database. Here, the most recent available SSURGO data (2008) was used to determine the saturated hydraulic conductivity of the different subcatchments. Other spatially variable properties include ground slope, percent imperviousness, impervious area depression storage, pervious area depression storage, suction head, initial moisture deficit, length of overland flow, and groundwater flow coefficient. These values were adopted from the original model.

In the original model, the pipe diameters varied from 3 to 9 feet, while the largest pipe diameter of 9 feet was used for all pipes in the present model. This modification was made to simplify the analysis and not require pipe sizing calculations for each resolution. The model results were not sensitive to using different or the largest diameter for all pipes. Spatially variable values for conduit slope and Manning’s roughness in the
original model were substituted with area-weighted average values of 0.013 ft/ft for conduit slope and 0.013 for Manning’s roughness. These simplifications did not significantly affect model output.

Using the above modifications, models were developed at various spatial resolutions using actual sewer networks. The coarsest resolution had four subcatchments, three pipes and three nodes. The intermediate resolution had 19 subcatchments, 20 pipes and 20 nodes. The densest resolution had 401 subcatchments, 402 pipes and 402 nodes. Next, models were also developed at corresponding resolutions using networks generated artificially based on a fractal algorithm. The next section will discuss in detail the generation of artificial sewer networks and describe the program that was developed as part of this study to generate the artificial sewer networks.

3.3 Generating Artificial Sewer Networks

Hydrologic simulation models are important tools for research and management of the urban environment. One of the primary data requirement for hydrologic simulations and key input for hydrologic models is the drainage network. Of course, the best network to use for simulations is the actual one. However, there are a number of cases where we may want to generate and use an artificial network:

- For many older cities, a consistent digital representation of the drainage network is not available. Generating such a representation from hard copy “as build” drawings can be very time consuming and often practically impossible for large areas. Using an artificial network could represent an alternative.
• For analysis of alternative future scenarios (planning), the “actual” sewer network is not known. Of course, it could be designed by engineers, but this would make alternative analysis more time consuming and expensive. An artificial network generated based on some planning variables (e.g. land use) could be used for such an analysis.

• Artificial networks can be used for research. The rapid generation of drainage networks at various densities can be useful for understanding the effect of scale on hydrologic simulations or the development of scaling laws. Much research on this problem has been done in rural hydrology (e.g., Wood et al. 1988; Song and James 1992).

• In some cases the best solution may be a hybrid combination of coarse-scale actual and fine-scale artificial network. This would allow the modeler to incorporate large-scale features, like pump stations or combined sewer regulators, where site-specific information cannot be neglected, and small-scale features, like individual gutters or driveways, where their exact location and properties are not as important. Also, the hybrid approach would allow the modeler to simulate at a consistent resolution, which would eliminate scaling effects introduced by applying non-linear models to heterogeneous data at varying scales (e.g., Beven 1989).

In rural hydrology, Geographic Information Systems (GIS) procedures exist that automatically generate watershed features (streams, watershed boundaries) and assemble them into a network for modeling (e.g., Hellweger and Maidment 1999; Maidment 2002). Procedures also exist that take an existing network and densify it (Hellweger 1997). However, in rural settings, the hydrology is mostly controlled by natural topography and the network can be generated from digital elevation model (DEM) raster grids. The question is, can a similar approach be developed for the urban environment, where the hydrology is not (directly) driven by the natural topography. Can
an automated approach of designing an urban drainage network be developed that produces the same results, in terms of peak flow, volume, quality parameters, etc. as a high-resolution manual set-up? The idea is to generate an artificial drainage network that may look different, but produces effectively the same results as the actual system.

For this purpose, the program for generating artificial sewer networks, called the Artificial Network Generator (ANGel) was developed as part of this study. The subsequent sections are organized in the following manner. Section 3.4 discusses the underlying methodology based on fractal geometry. Section 3.5 presents the all the input options available to the user in the form interface to ANGel. Section 3.6 discusses the program structure in ANGel. Section 3.7 explains the definitions of the SWMM input parameters that are generated using ANGel, and discusses the approach used to estimate these input parameters for the sub-basin area used in this study. Section 3.8 describes how ANGel output is imported into SWMM.

3.4 Methodology (Fractal Geometry)

Fractals are geometric shapes that are self-similar over a wide range of scales, and fractal trees are a class of fractals consisting of a network of connecting lines. Examples of natural networks that have been associated with fractal trees include tree branches, blood arteries and natural drainage systems. Although fractals were not introduced by Mandelbrot until 1967, Horton’s (1945) empirical work indicates that the bifurcation ratio (ratio of number of streams in successive stream generations) and the length-order ratio (ratio of mean length of streams in successive generations) of natural drainage networks are nearly constant across scales. As a result, fractals have been the subject of significant interest in rural hydrology (e.g., Hjelmfelt 1988; Tarboton et al.)
A fractal tree could be used to represent an urban drainage network. However, there are numerous types of trees, many having properties inappropriate for the urban environment. Many binary trees are not “space-filling” and are “self-touching” (inconsistent with the concept of a dendritic network). A special type of binary tree, the “H”-tree (bifurcation ratio = √2, angle = 90°), is space-filling and not self-touching. Binary trees with side branches can be space-filling. However, the binary structure results in intersections of three lines, whereas in the urban environment above- and below-ground flow paths often follow roads that often form intersections of four lines. It may be that the most appropriate fractal tree is the one referred to by Turcotte and Newman (1996) as “Tokunaga” tree, shown in Figure 3.2. This fractal tree is space-filling and not self-touching. Note that, although similar, the fractal tree presented in Figure 3.2 is different from the “Peano” tree discussed by Marani et al. (1991) and Tarboton (1996).

Figure 3.2. Construction of a “Tokunaga” fractal tree (from Turcotte and Newman, 1996).
The Tokunaga fractal tree is constructed from a straight line (initiator) by (1) splitting the initiator and (2) adding three lines at 90°, 180°, 270° angles at the end point of the initiator. The procedure is repeated for each subsequent generation for each line, with the exception that no redundant lines will be added.

The geometry of a network tree can be summarized using a number of parameters, including Horton’s bifurcation ratio:

\[ R_b = \frac{N_i}{N_{i+1}} \]  \hspace{1cm} (1)

where, \( N_i \) is the number of streams of order \( i \), where “order” refers to the Strahler stream ordering system. The bifurcation ratio is the ratio of the number of streams of one order to the number of streams of the next higher order. For two generations of the Tokunaga fractal tree, \( R_b = 3/1 \), and for many generations \( R_b \to 4 \). Horton also introduced the length-order ratio:

\[ R_r = \frac{r_{i+1}}{r_i} \]  \hspace{1cm} (2)

where \( r_i \) is the mean length of streams of order \( i \). The Tokunaga fractal tree has a constant \( R_r = 2 \). The fractal dimension is (Turcotte 1997):

\[ D = \frac{\ln\left(\frac{N_i}{N_{i+1}}\right)}{\ln\left(\frac{r_{i+1}}{r_i}\right)} \]  \hspace{1cm} (3)
The substitution of (1) and (2) gives the fractal dimension of a drainage network in terms of $R_b$ and $R_r$:

$$D = \frac{\ln R_b}{\ln R_r}$$  \hspace{1cm} (4)

For two generations, the Tokunaga fractal tree has $D = \ln(3)/\ln(2) = 1.58$, and for many generations $D \to 2$.

### 3.5 User Input Options

The program, called Artificial Network Generator (ANGel), was written in Visual Basic for Applications (VBA) for ArcGIS 9.2. The program includes options such as specifying a user-defined starting network, using an existing shapefile and existing drainage areas for analysis, combining short pipes or segmenting longer pipes in a network, adding generations to a network, randomizing the network, specifying user-specified input for different hydrologic parameters of subcatchments such as ground slope, percent imperviousness, soil properties, etc, and the option to specify the average slope of pipes in the network. All user input specified on the form in ANGel is shown in Figure 3.3.
The ANGel program appears as a toolbar called “Fractal” on the ArcMap application of ArcGIS. When this toolbar is enabled, a command button appears on the map document called “Run ANGel”. The above user input form is activated by clicking on this button. The user should not have any layers on the map document that have a missing reference (i.e., any layer that has been deleted from its source location), otherwise the input window fails to activate.

Figure 3.3. User input window of the Artificial Network Generator (ANGel), version 1.2.
3.5.1 Starting Network

The program works by taking a network (the “starting network”), also called the initiator (blue line in Figure 3.4), and expanding it. The user has the option of specifying the starting network as a straight line by typing the coordinates of the start and end points into text boxes in the form, or by clicking on the map. The other option is to use an existing shapefile as the starting network. Figure 3.4 shows two artificial networks. The network in Figure 3a was created using user-supplied X/Y coordinates. The other one was created by expanding an existing network. Note that ANGel uses the convention that lines are directed from downstream to upstream, so an existing shapefile will have to conform to that.

Figure 3.4. (a) A fractal network with user-specified starting network, (b) A fractal network with existing shapefile as starting network.

3.5.2 Existing Drainage Areas

The user also has the option to use existing drainage areas or subcatchments (in the form of GIS shapefiles) that are associated with the drainage network (pipes and nodes) for a catchment area. This means that an existing drainage network and its corresponding drainage areas can be analyzed using ANGel. This is particularly useful when hydrologic properties need to be re-evaluated with respect to changes in input data (e.g. land use) without changing the resolution of the input data. The GIS shapefile for the existing subcatchments needs to be added to the active map document before it
is specified by the user. This means that for every drainage network, the user has two options. He may use the existing drainage areas associated with that network, or, if this is not available, he may generate the subcatchments artificially using ANGel (based on the Voronoi polygon function, discussed in detail in previous chapter). Figure 3.5 illustrates both these options. Figures 3.5a and 3.5b show the program output, respectively when existing drainage areas associated with an existing network are used and when drainage areas are artificially generated using ANGel.

3.5.3 Combining Pipes in Drainage Network

The present version of the program has the option of merging smaller pipes in an existing dense network. This implies that this option is usually used when the user chooses an existing drainage network to calculate hydrologic attributes or would like to generate drainage areas corresponding to an existing network. This option appears as the check box “Combine Lines in Existing Network” in the ANGel form. This could be particularly useful in cases where the user is not certain if these smaller pipes actually
exist in the drainage network or merely are a consequence of poor digitization of the actual pipes in GIS. This option of combining pipes in a network also has the advantage of avoiding flow instabilities resulting from very small pipes in the network. The function used to execute this option is designed on the basis of merging lines (pipes) in the network that have one line (pipe) upstream of it with this upstream line (pipe). However, the user should be careful in using this option for networks that are not as dense, or which may have some longer conduits upstream in the network. In such cases merging lines may result in longer pipes that are no longer representative of the pipes in the actual drainage network. Figure 3.6 shows a part of the drainage network (magnified) to illustrate how the smaller pipes in the existing network (Figure 3.6a) may be merged to form longer pipes in the network (Figure 3.6b).

Figure 3.6. (a) Original network with shorter pipes (b) Modified network with shorter pipes merged. Note the shorter pipe segments in (a) have been merged to form longer pipes in (b). Note that arrow heads point to the upstream direction.
3.5.4 Splitting Pipes in Drainage Network

Contrary to combining smaller pipes in a network, the user may also choose to split longer pipes in the drainage network. This option is particularly useful in two cases. First, if the user wishes to discretize a low-resolution model with longer pipes, fewer drainage nodes and larger drainage areas into a high-resolution model with smaller pipes, their corresponding drainage nodes and smaller drainage areas. Second, if merging several smaller pipes in a network results in unreasonably longer pipes, then this feature may be used to split these longer pipes into smaller pipes of equal lengths.

This option appears as the check box “Split Distance”, which allows the user to divide a long pipe segment into two pipes of equal lengths. Figure 3.7a shows a segment of a drainage network at a coarser resolution with longer pipes, fewer nodes and larger drainage areas. Figure 3.7b shows the same network when discretized into a higher resolution with shorter pipes, more drainage nodes and smaller drainage areas.

Figure 3.7. (a) Coarser resolution network with longer pipes (b) Finer resolution network with smaller pipes, generated using the “Split Distance” option in ANGel. Note that arrow heads point to the upstream
Along with the check box option “Split Distance” used to split longer pipes, the ANGel form also has the option of a text box to specify the desired maximum length of pipes in the system, and if this pipe length is exceeded the pipes will be split into two pipes of equal length. In other words, this input distance will make sure that no pipes in the network can be longer than this specified length. Therefore, in some cases it may be more meaningful to use this option along with the option of “Combine Lines in Existing Network”. For example, if the user chooses to merge smaller upstream pipes in the network, which leads to some longer pipes that are no longer representative of the actual network, he may check on the “Split Distance” option and specify that no pipes in the resulting network may be longer than say, “500 ft”. This means that all pipes that are longer than 500 ft will be split into two pipes of equal length. Figure 3.8a shows a segment of a drainage network with shorter pipes. When these smaller pipes were merged using the “Combine Lines in Existing Network” option, it resulted in pipes that were much longer than 500 ft. In this case, each of these resultant longer pipes was split into two smaller pipes of equal length (Figure 3.8b).

![Figure 3.8. (a) Original network with smaller pipes (b) Original pipes that were merged to form longer pipes, and were subsequently split into shorter pipes of equal lengths. Note that arrow heads point to the upstream direction.](image)
### 3.5.5 Fractal Type

The current version of the program supports only the Tokunaga fractal tree as network shape. The program could be extended in the future to include other algorithms, such as the H-Tree and Peano fractals as well as “fishbone” (NFB) shapes.

### 3.5.6 Number of Generations to Add

The user specifies the number of generations to add to the starting network. The starting network is designated as generation 1. If the user specifies to add two generations to the starting network, a three-generation network will be created. The program also accepts zero as input, for when the user only wants to calculate statistics (e.g. stream order, average flow length, Horton’s ratio, length-order ratio, drainage density) of an existing network. Figure 3.9 shows three artificial networks. The network in Figure 3.9a was created by adding no generation to an existing network, while the one in Figure 3.9b was created by adding four generations to a user-specified starting network (yellow line). The network in Figure 3.9c is an example of a relatively complex network. It was created by adding five generations to a user-specified starting network (cyan line), and also by introducing irregularity in the network. Note that in ANGel by default the streams in the fractal network are colored by their Strahler stream order.

![Figure 3.9](image)

Figure 3.9. (a) A fractal network generated by adding zero generations to an existing shapefile. (b) A fractal network generated by adding four generations to a starting network, (c) A complex fractal network generated by adding five generations to a starting network. Note that (c) is also irregular in shape.
3.5.7 Advanced Options

The user specifies a “spatial tolerance” distance, which describes the spatial accuracy used by the program. In general, the value should be based on the spatial accuracy of the input shapefiles. Note that all input and output distances are in map units. That means if the map distance units are in meters, the drainage density is in m/m². The program uses the spatial tolerance in a number of ways:

- The program determines the existence of connections based on a “spatial tolerance” distance. That means if two points are within a distance of less than the spatial tolerance, they will be considered to be connected.

- The program also identifies the existence of intersections amongst the lines (which can happen when generating an irregular network or expanding an existing network) and shortens one of these intersecting lines based on the “spatial tolerance” distance. If a line intersects another line (that is neither its downstream line nor one of its upstream lines) the program will shorten one of the lines (the one of the higher generation).

- The spatial tolerance is also used to identify and remove short lines from the network. The short lines are usually the result of clipping a network to the extents of a catchment area.

- The program uses a buffer for clipping the network to the boundary of the catchment area. The spatial tolerance value input in the form also determines the buffer distance from the boundary of the catchment area at which the network will be clipped.
The user can specify a catchment area to clip the artificial network to. The program automatically populates the pull-down menu “Clip to Basin Area” with the polygon shapefiles in the table of contents. Figure 3.10 shows a fractal network before and after it has been clipped to a catchment area boundary. The program calculates and assigns the stream IDs and Strahler stream order of the lines and traces the downstream IDs of the lines both before and after the clipping operation. This is essential to establish the correct topological relationships between the lines, which may have changed as a result of clipping. Clipping can create “orphans”, lines or entire portions of the network, which are cut off from the main network. Those lines are identified and removed.

![Figure 3.10. (a) A fractal network before (a) and after (b) clipping to the extents of a catchment area.](image)

Networks generated using strict fractal algorithms are very regular, but actual networks are seldom so. Therefore, the program can generate an irregular network. The user specifies the degree of irregularity by entering a percentage in the “Irregular” option. Figure 3.11 shows two artificial networks with the same starting network and number of generations. One of them was generated without the irregular option, while the other one was generated with an irregularity of 30 percent. The user can generate the same irregular network every time by inputting the same positive integer value (e.g.,
1, 2, etc.) in the “Seed” option, which is the seed value used by the random number generator. A different network is generated by setting the “Seed” option to ‘-1’, which means the program will use the system date as seed value.

![Figure 3.11. (a) A regular fractal network. (b) An irregular network with 30 percent irregularity.](image)

### 3.5.8 Input Options for SWMM

ANGel is designed such that the it can generate all input parameters required to run the Stormwater Management Model, SWMM (Rossman 2005). In this version, the user has the option to input raster datasets for subcatchment hydrologic parameters. These appear as input boxes on the user form in ANGel, each with a pull down menu that has all the raster grids available to the user. The user can select the appropriate grid dataset by clicking on the right option from the menu. The form has input options for the following subcatchment hydrologic parameters: length of overland flow (LOF, ft), average ground slope (%), percent imperviousness, impervious area depression storage (in.), pervious area depression storage (in.), soil suction head (in.), saturated hydraulic conductivity of soil (in./h), soil initial moisture deficit, and groundwater flow coefficient.

Based on the respective input raster grids for each of the above parameters, ANGel can assign the corresponding property to subcatchment areas using the Zonal Statistics
function based on the mean statistic. The details of the Zonal Statistics function will be discussed in the next section on program structure.

The subcatchment areas, which serve as the zone dataset for the Zonal Statistics function could be existing drainage areas associated with an actual network (specified in the input box “Existing Areas”), or they could be areas that are generated artificially using ANGel. For a particular sub-basin, the user may have the input raster data at any spatial resolution, and the program is able to assign corresponding hydrologic parameters (based on this input data) to subcatchment areas at a different spatial resolution. This means that heterogeneity of properties among subcatchments will be more if the resolution of the input data is higher than the resolution of the subcatchment areas. On the other hand, the heterogeneity of properties among subcatchments will be lower if the input data is at a lower resolution than the resolution of the subcatchment areas. In Figure 3.12a, raster grids for percent imperviousness were overlaid on subcatchment areas at a lower resolution, while in Figure 3.12b, the same raster grids were overlaid on subcatchment areas at a higher resolution.

![Figure 3.12. Input data for percent imperviousness as raster grids (shown as graded colors) with subcatchments overlaid at (a) lower resolution than data; (b) higher resolution than data.](image)
The output of the Zonal Statistics function (using the mean statistic), applied on raster dataset for each input parameter and zone areas, appears as the corresponding attribute for each subcatchment area of the area shapefile generated by ANGel. Other than the hydrologic input parameters required to run SWMM, ANGel also calculates hydraulic parameters for the drainage network (e.g., conduit lengths). A new hydraulic parameter that is calculated by the present version of ANGel is the invert elevation of the nodes in the network. The user can input an average slope of the pipes in the network, which is used by the program along with the respective conduit lengths to assign invert elevations of all the drainage nodes in the network (assuming the outfall point has zero elevation).

3.5.9 Output

Program output consists of point, line and area shapefiles, and a text file listing statistics. Future plans are to generate output in the form required by sewer models (Rossman 2005). The line shapefile has a polyline for each sewer line. The point shapefile has a point at the upstream end of every line. The area shapefile has drainage area polygons for each line and is generated by using the “ConvertToVoronoiRegions” method of ArcGIS, which writes the set of Voronoi polygons for each node (or point in the point shapefile) to a feature class. The Voronoi region (also known as Thiessen or proximal polygon) encloses an area that is closer to the source node than to any other node in the triangulation. Table 3.1 lists the attributes of the output point, line and area shapefiles. The area, node and line shapefilea are added to the table of contents on the map and the lines are and symbolized using the “Unique Values” option on the “StrOrder” (Strahler stream order) field. Box 1 shows an example of an output statistics text file.
<table>
<thead>
<tr>
<th>Output ShapeFile</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Point</strong></td>
<td>ID</td>
<td>A unique integer ID for each node in the network.</td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td>An alphanumeric ID for each node, e.g. node ID 1 has node name N_1.</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X co-ordinate of the node location on the map.</td>
</tr>
<tr>
<td></td>
<td>Y</td>
<td>Y co-ordinate of the node location on the map.</td>
</tr>
<tr>
<td></td>
<td>DNode_Name</td>
<td>Name of the immediate downstream node.</td>
</tr>
<tr>
<td></td>
<td>NUp_Nodes</td>
<td>Number of immediate upstream nodes.</td>
</tr>
<tr>
<td></td>
<td>DLine_Len</td>
<td>Length of the conduit downstream of the node, in feet.</td>
</tr>
<tr>
<td></td>
<td>Inv_elev</td>
<td>Invert elevation of the node, in feet.</td>
</tr>
<tr>
<td><strong>Line</strong></td>
<td>ID</td>
<td>A unique integer ID for each pipe in the network.</td>
</tr>
<tr>
<td></td>
<td>In_Node</td>
<td>Name of the inlet node of the pipe.</td>
</tr>
<tr>
<td></td>
<td>Out_Node</td>
<td>Name of the outlet node of the pipe.</td>
</tr>
<tr>
<td></td>
<td>NUpstr</td>
<td>The total number of pipes upstream of the pipe in the network.</td>
</tr>
<tr>
<td></td>
<td>Generation</td>
<td>The generation number of the line in the network (in terms of fractal geometry).</td>
</tr>
<tr>
<td></td>
<td>DownID</td>
<td>The ID of the pipe immediately downstream of it.</td>
</tr>
<tr>
<td></td>
<td>Outlet</td>
<td>A Boolean which indicates whether the line has the outlet immediately downstream of it or not.</td>
</tr>
<tr>
<td></td>
<td>Temp</td>
<td>A temporary variable assigned to the line during the network generation.</td>
</tr>
<tr>
<td></td>
<td>Orphan</td>
<td>A Boolean which indicates whether the line is a straggler (has no upstream line) or not.</td>
</tr>
<tr>
<td></td>
<td>Length</td>
<td>The length of the line, in meters.</td>
</tr>
<tr>
<td></td>
<td>Length_ft</td>
<td>The length of the pipe, in feet.</td>
</tr>
<tr>
<td></td>
<td>FlowLength</td>
<td>The total length of the line to the outlet of the network, in meters.</td>
</tr>
<tr>
<td></td>
<td>StrOrder</td>
<td>The Strahler stream order of the network (in terms of fractal geometry).</td>
</tr>
<tr>
<td></td>
<td>StrID</td>
<td>The stream ID of the line in the network.</td>
</tr>
</tbody>
</table>
Table 3.1. Attributes of the output point, line and area shapefiles generated by ANGel, version 1.2 (contd.)

<table>
<thead>
<tr>
<th>Output ShapeFile</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>ID</td>
<td>A unique integer ID for each subcatchment area in the drainage network.</td>
</tr>
<tr>
<td></td>
<td>Area</td>
<td>The drainage area of the subcatchment, in square meters.</td>
</tr>
<tr>
<td></td>
<td>Area_ac</td>
<td>The drainage area of the subcatchment, in acres.</td>
</tr>
<tr>
<td></td>
<td>NodeIndex</td>
<td>A field generated automatically by the ‘ConvertToVoronoiRegions’ function.</td>
</tr>
<tr>
<td></td>
<td>TagValue</td>
<td>A field generated automatically by the ‘ConvertToVoronoiRegions’ function.</td>
</tr>
<tr>
<td></td>
<td>Outlet</td>
<td>Name of drainage node that receives subcatchment runoff or overland flow.</td>
</tr>
<tr>
<td></td>
<td>LOF</td>
<td>Length of overland flow for the subcatchment, in feet.</td>
</tr>
<tr>
<td></td>
<td>Width_ft</td>
<td>Subcatchment width parameter, in feet.</td>
</tr>
<tr>
<td></td>
<td>WSLOPE</td>
<td>Average surface slope of the subcatchment, in %.</td>
</tr>
<tr>
<td></td>
<td>IMPERV</td>
<td>Percent of total impervious area of the subcatchment.</td>
</tr>
<tr>
<td></td>
<td>WSTORE1</td>
<td>Depth of depression storage on the impervious area of the subcatchment, in inches.</td>
</tr>
<tr>
<td></td>
<td>WSTORE2</td>
<td>Depth of depression storage on the pervious area of the subcatchment, in inches.</td>
</tr>
<tr>
<td></td>
<td>INFIL1</td>
<td>Soil capillary suction head of the subcatchment, in inches.</td>
</tr>
<tr>
<td></td>
<td>INFIL2</td>
<td>Soil saturated hydraulic conductivity of the subcatchment, in in./h.</td>
</tr>
<tr>
<td></td>
<td>INFIL3</td>
<td>Initial soil moisture deficit fraction for the subcatchment</td>
</tr>
<tr>
<td>Area</td>
<td>A1</td>
<td>Groundwater flow coefficient.</td>
</tr>
</tbody>
</table>


Box 1: An example output of the Stats text file

In the program the default path of all the output files is set to the current directory in which the program is saved. The default filename for the line fractal always begins with ‘L’ followed by ‘Fractal1’, say. The default filenames for the point fractal and the area fractal shapefiles begin with ‘P’ and ‘A’, respectively followed by the same ‘Fractal1’, say. The default filename of the statistics text file is ‘Fractal1.txt’. These default names get updated every time the program is run with ‘LFractal2’, ‘PFractal2’, ‘AFractal2’, ‘Fractal2’, ‘LFractal3’, ‘PFractal3’, ‘AFractal3’, ‘Fractal3’, and so on. This ensures that all the output files have new names assigned to them every time that ANGel is run.
3.6 Program Structure

This section describes the main structure (program flow). The first step in the program is to develop the initiator. If the user selects to use an existing network, the program will read the specified shapefile. It will then calculate topology (connectivity) amongst the lines. Then the outlet is found as the line without any downstream lines. Junctions are snapped, which is done for aesthetic reasons. Lines that connect without additional tributaries entering at the junction are combined (dissolved) into one line.

However, if the options “Combine Lines in Existing Network” and “Split Distance” are checked, then the program flow is modified as shown in Figure 3.13. In this case, lines (or pipes) that have one line upstream of it are merged to form one pipe using the function “DissolveLines”. The new lengths for each of the pipes are calculated and compared with the length specified in the “Split Distance” input box, which is a user-input and allows the user to specify the maximum allowable length of pipes in the network. If the lengths of the merged lines are greater than this maximum length, they are split into two lines of equal lengths, and the lengths are recalculated. If the line lengths are less than the maximum value, their lengths are kept unchanged.
Figure 3.13. Program flow in the initial stage when existing drainage network is used.
The other significant new feature in the present version of the program is that it can use existing drainage areas associated with an existing drainage network. Therefore, if the user chooses to analyze an existing drainage network in ANGel, without adding any new generations to it, he may either artificially generate the drainage areas or use the existing drainage areas associated with the network. These options are incorporated in the function that creates the output areas shapefile (function WriteAFractal(TempKey as String)) and the flow of code within this function is highlighted in Figure 3.14. When artificial areas are generated using ANGel, the following steps are followed. First, the drainage nodes shapefile is added to a new Triangulated Irregular Network (TIN), and this TIN is assigned to a TinNodeCollection object. Now a thiessen polygon is delineated for each drainage node using the function “ConvertToVoronoiRegions”, and these polygon areas are clipped to the extents of the main study area. Finally, they are stored in the output areas shapefile. When the option “Use Existing Areas” is used, two feature cursors are created: one for the existing areas shapefile and the other for the areas shapefile that will be written by ANGel as output. Using the respective cursors, each feature from the existing areas shapefile is copied to the output shapefile, and output areas are calculated in acres.
Use Existing Drainage Network

If Use Existing Areas

- Define Cursor1 for Existing Areas shapefile
- Define Cursor2 for AFractal shapefile

- Set Cursor1 to Old (1st) Feature in Existing Areas shapefile
- Set Cursor2 to New (1st) Feature in AFractal shapefile

Do Until Old Feature is Nothing

- Copy Old Feature to New Feature
- Calculate Area of New Feature
- Store New Feature
- Update Cursor1 to next Old Feature

If Use Existing Areas True

- Store New Feature

Create new Tin

Add PFractal to Tin

Assign Tin to TinNodeCollection

Create Voronoi Polygons for each Node

Store Polygons to AFractal shapefile

Assign other attributes to AFractal shapefile

Create new Tin

Add PFractal to Tin

Assign Tin to TinNodeCollection

Create Voronoi Polygons for each Node

Store Polygons to AFractal shapefile

Assign other attributes to AFractal shapefile

Figure 3.14. Program flow when existing drainage areas are used for analysis or when they are generated artificially.
Once the initiator has been developed, the network is expanded based on the number of generations input by the user. The flow chart of the program to add successive generations to the initiator is illustrated in Figure 3.13. An expansion consists of two basic steps: (1) splitting the line and (2) adding three lines at 90°, 180°, 270° angles at the end point of the line. When adding the three lines, the percent irregularity, if input by the user, is taken into account. Note that the lengths of these extensions in an irregular fractal will not be the same, unlike a regular fractal, and will depend on the percent irregularity and the seed value input by the user. The procedure is repeated for each subsequent generation for each line, with the exception that no redundant lines will be added. After the expansion is complete, the program again calculates the topology (connectivity) amongst the lines. Next, the ‘unwanted intersections’ amongst the lines are identified. Typically, expansion for any irregular network and even a regular network developed from an existing network will almost always result in such ‘unwanted intersections’ of the lines. It is okay for two lines to intersect if either of them is upstream or downstream of each other, or if they intersect at a junction (i.e., have the same downstream ID). Any other intersection in the network apart from these will be classified as ‘unwanted intersection’. All the lines are checked for intersections amongst themselves and unwanted intersections are marked. Then the higher-generation line in these unwanted intersections is shortened so that it comes within a distance of twenty times the spatial tolerance (specified in the input form) of the other. At this stage, the lengths of each the lines in the network are calculated, and the program removes all lines in the network whose lengths are less than twenty times the spatial tolerance (specified in the input form).

Next, if the user chooses to clip the network to the extent of a catchment area, a new shapefile (temporary version “a”) with line features is created. Then, a buffer area
shapefile is created around the catchment boundary at a distance of twenty times the spatial tolerance (specified in the input form) from the boundary. Now, if no generations are added to the starting network, the program reads the network shapefile (temporary version “a”). If the number of generations added to the starting network is greater than zero, the network shapefile (temporary version “a”) is clipped to the catchment area and a new line shapefile (temporary version “b”) of the network is created. The program then reads this new shapefile (temporary version “b”).

Irrespective of whether the network has been clipped or not, the next step in the program is to remove all lines in the network, which are shorter in length than twenty times the spatial tolerance (specified in the input form). Also, at this stage irrespective of clipping or not, the topology (connectivity) amongst the lines is recalculated to establish any changes in connectivity that might have occurred as a result of clipping.

The clipping operation often results in ‘stragglers’ or ‘orphans’, which are lines that have been cut off from the network. These are now identified and removed by the program. The lengths and the flow lengths (length to the outlet) of each of the lines in the network are calculated. The Strahler stream order and stream ID of each line are established.

Next, a node is created at the upstream end of each line. This results in a point fractal from which a point shapefile is created. Each of these nodes in the point shapefile will have a drainage area associated with it. The program then generates the area shapefile in which Voronoi regions or proximal polygons are created for each node, enclosing an area that is closer to the source node than to any other node in the point shapefile. At this stage, the final version of the network (line shapefile) is created by writing all the attributes to it like the stream lengths and flow lengths, the stream IDs and their Strahler stream order, and the drainage area associated with each line. Next the
point, line and area shapefiles are directly added to the map if the user has selected the respective options on the form. After adding the network to the map, the program colors the lines in the network by their Strahler stream order. Finally, the statistics text file is generated if the user selects to do so. This text file summarizes the basic line (length, flow length and area), stream (stream length), hydrology (bifurcation ratio, length-order ratio, drainage density), and fractal (fractal dimension) statistics of the network.

Figure 3.15. Program flow to add generations to the starting network.
Figure 3.15. Program flow to add generations to the starting network (contd.).
The present version of ANGel can assign attributes to nodes, pipes and areas shapefile that are needed to run a SWMM simulation. The advantage of this feature is that these output shapefiles can be imported to SWMM and simulations can be performed with different drainage networks generated under different scenarios. The functions related to these modifications are all included in the main code after the point (PFractal), area (AFractal) and line (LFractal) shapefiles have been created. This section of the program flow is presented in the flow chart in Figure 3.15.

First, the attributes in the drainage lines shapefile are updated. For this a feature cursor is defined, respectively for the node and the line shapefiles. The cursor to the point feature class is looped through all the features for each feature in the line feature class. By convention the lines point upstream in the shapefile. However, in SWMM the flow direction always points towards the outfall point of the model. Therefore, the node at the starting point of the line will be its outlet node according to SWMM, while the node at the end point of the line will be its inlet node according to SWMM. The distance between the starting point for each line feature in the line shapefile and each node in the point shapefile is measured. When this distance is less than the allowable spatial tolerance value for a particular node, this node is assigned as the inlet node for that pipe. A similar approach applied to the end point for each line feature to determine the outlet node for the pipe.

Next, the invert elevations of the nodes are calculated based on the input for average pipe slope specified in the “Pipe Slope” input box in the form. The default value used is 0.013. At first, for each node in the point shapefile, the length of the line (in feet) downstream of it (DLine_Len) and the name of downstream node (DNode_Name) are determined. Also, for each node, the total numbers of upstream nodes (NUp_Nodes) are determined. The invert elevation of the outfall point is assumed to be zero. Then the
invert elevation for each node in the point shapefile is calculated by adding the elevation of the downstream node to the product of the average pipe slope and the length of the line downstream of that node.

Finally, the SWMM attributes for the drainage areas in the area shapefile are calculated. The outlet for each subcatchment is the node that is contained by that subcatchment. The other hydrologic attributes of the subcatchments are calculated using the Zonal statistics function on the respective input raster dataset. The Zonal Statistics function can be used to compute statistics for each zone of a zone dataset based on the information in a value raster (McCoy and Johnston 2002). While the zone dataset can be feature or raster data, the value raster must be a raster dataset (McCoy and Johnston 2002). The Zonal Statistics function is computed for each zone, and a single output value is determined for every zone. The statistics functions that can be used include maximum, minimum, mean, median, range, standard deviation and sum among other functions. To generate SWMM attributes, either the existing drainage areas in a network or the artificial areas generated using ANGel are used as the zone dataset (vector data). Depending on which raster input option is checked on the ANGel form, the zonal statistics function is executed with each of the input datasets and the area shapefile. The two input parameters used with the ZonalStatistics function are the name of the input raster dataset and the name of corresponding SWMM attribute that will be written to the output area shapefile. The options for input raster datasets are length of overland flow (LOF), ground slope, percent imperviousness, pervious and impervious area depression storage, soil suction head, saturated hydraulic conductivity, initial moisture deficit and groundwater flow co-efficient. When the option for LOF is checked, the function to calculate the subcatchment width parameter (CalcWidth) is also called. The width parameter for a subcatchment is calculated by dividing the subcatchment area by its
corresponding LOF. It is to be noted that when any of the SWMM input options is checked, the options to add the drainage areas and nodes shapefiles to the map document are automatically activated. Areas are calculated in acres.

![Flowchart Diagram](image)

Figure 3.16. Program flow to generate SWMM attributes for the nodes, pipes and areas shapefiles.
3.7 Description and estimation of SWMM input parameters

This section explains the definitions of the SWMM input parameters that are generated using ANGel, and discusses the approach used to estimate these input parameters for the study area Faneuil Brook sub-basin of the lower Charles River watershed. First, the definitions of the hydrologic parameters in SWMM are described below.

3.7.1 Description of SWMM parameters

*Length of overland flow (LOF) and width parameter: LOF for a subcatchment can be defined as the length of the flow path from the hydraulically most distant point of the subcatchment to its outlet (TR-55 1986). Overland routing of surface runoff from a subcatchment to its inlet node or pipe is a function of its LOF. One or many such flow paths can be conjectured for a subcatchment or a self-consistent method can be used to determine LOF (James et al. 2005). The subcatchment width parameter is a derived parameter and can be calculated by dividing subcatchment area by its LOF. Input LOF values appear as LOF and *Width_ft in the shapefile for drainage areas.*

*Subcatchment slope:* This is the average slope along the pathway of overland flow to inlet (node) locations. This could be determined as the elevation difference divided by the length of flow for a simple geometry. For subcatchments with complex geometries, several possible overland flow paths can be delineated and a path-length weighted average subcatchment slope may be determined. This input appears as *Ground Slope* in ANGel and the corresponding attribute is *WSLOPE* in the areas shapefile.

*Percent imperviousness:* The sum of the areas of impervious surfaces such as roads, sidewalks, driveways, parking lots and rooftops that are expressed as a percentage of
the total drainage area is called percent imperviousness (Viessman and Lewis 2003). SWMM distinguishes between areas that are hydraulically connected to the drainage system (effective impervious area) and those that drain onto adjacent pervious areas (James et al. 2005). This input appears as % Imperv in ANGel, and the corresponding attribute is IMPERV in the areas shapefile.

*Impervious and pervious area depression storage:* Depression storage is a volume of water that must be filled on both impervious and pervious areas before surface runoff occurs (Viessman and Lewis 2003). It represents a loss or “initial abstraction” due to surface ponding, interception and evaporation. In SWMM, a parameter PCTZER can be assigned to subcatchments, which represents the percent of impervious area assigned zero depression storage to promote immediate runoff. This value may be same or different for different subcatchments. Inputs for pervious and impervious area depression storage appear as IDS and PDS in ANGel, and the corresponding attributes are WSTORE1 and WSTORE2, respectively in the areas shapefile.

The next three parameters are related to the Green-Ampt infiltration model.

*Soil suction head:* This represents the average capillary suction head at the wetting front of water. This is a relatively difficult parameter to estimate, and this data may be obtained from soil moisture versus soil tension data (James et al. 2005). Input for suction head appears as SUCT in ANGel and the corresponding attribute is INFIL1 in the areas shapefile.

*Saturated hydraulic conductivity (Ks):* This is a measure of soil permeability, and represents the rate of water movement through saturated media. The range of saturated hydraulic conductivity values for different soils is of the order of a few tenths of an inch per hour. These values for different soil types can be obtained from the Soil Survey
Geographic (SSURGO) Database (USDA 2008). Input for saturated hydraulic conductivity appears as HYDCON in ANGel and the corresponding attribute is INFIL2 in the area shapefile.

*Initial Moisture Deficit (IMD):* This is defined as the fraction difference between soil porosity and actual moisture content (Viessman and Lewis 2003), and is the most sensitive of the infiltration parameters to estimate runoff from pervious areas (James et al. 2005). Sandy soils typically have lower porosities than clay soils, but drain to lower moisture content between storms. Therefore IMD values for dry antecedent conditions are higher for sandy soils compared to clay soils. James et al. (2005) provides a good initial estimate of IMD values for different soil types under dry antecedent conditions. Input for IMD appears as SMDMAX in ANGel and the corresponding attribute is INFIL3 in the area shapefile.

*Groundwater flow coefficient:* This is the multiplier of distance between actual and threshold groundwater levels in the equation to predict groundwater flow between the aquifer underneath a subcatchment and the outlet node of the drainage system. This input appears as A1 both in ANGel and in the output area shapefile.

### 3.7.2 Estimation of SWMM input parameters

The general approach used to estimate the raster datasets for each of the above input parameters, except soil saturated hydraulic conductivity, was the same. These values were adopted from the model set-up developed by Zarriello and Barlow (2002) for the Faneuil Brook sub-basin. This original model was subdivided into eighteen subcatchments and also included a pond as a storage element. This original model was modified to remove the pond, which was replaced by a subcatchment of equal area. Area-weighted average values for each of the parameters obtained from the other
eighteen subcatchments were used as parameter values for this substitute subcatchment. For example, if we consider the parameter subcatchment slope, its average value is 8.7%, which was assigned to the substitute subcatchment. Next, the conversion tool “Feature to Raster” from Arc Toolbox was used to convert the vector slope data for these 19 subcatchments into an output raster dataset. This tool has four input parameters: input features, field, output raster and output cell size. The shapefile with these 19 subcatchments served as the “input features”, attribute for which the raster data was being generated is the “field” (e.g. WSLOPE in this case), the raster dataset created is the “output raster”, and usually the default value was chosen for the “output cell size”. Therefore, a raster dataset of subcatchment slope was created for the Faneuil Brook sub-basin (Figure 3.17a), which was used as the input raster for the “Ground Slope” option in ANGel. Using the Zonal Statistics function in ANGel, this raster dataset was overlaid on an area shapefile at any resolution (containing either actual or artificial areas) and a subcatchment slope was assigned to each subcatchment area (in the shapefile) at that resolution (Figure 3.17b). A similar approach was followed for all other input raster datasets except saturated hydraulic conductivity. A list of all input raster datasets along with their respective minimum, maximum, mean and standard deviation values have been presented in Table 3.2.
Figure 3.17. (a) Input raster dataset for subcatchment slope (in %) derived from initial model set up by Zariello and Barlow (2002); (b) raster grids for slope overlaid on drainage areas (corresponding to an artificial network) to calculate slope for subcatchments using the Zonal Statistics function in ANGel.

Figure 3.18. (a) Input raster dataset for saturated hydraulic conductivity derived from SSURGO data for Norfolk-Suffolk and Middlesex counties; (b) raster grids for saturated hydraulic conductivity overlaid on drainage areas (corresponding to actual network at a coarse resolution) to calculate $K_s$ for subcatchments using the Zonal Statistics function in ANGel.
Table 3.2. List of input raster datasets used for the study area Faneuil Brook sub-basin to calculate SWWM attributes for drainage areas at different resolutions using the Zonal Statistics function in ANGel.

<table>
<thead>
<tr>
<th>Input Raster Dataset</th>
<th>Maximum</th>
<th>Minimum</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of overland flow, LOF (ft)</td>
<td>5221.43</td>
<td>175.54</td>
<td>2661.43</td>
<td>1891.07</td>
</tr>
<tr>
<td>Subcatchment slope (%)</td>
<td>16.5</td>
<td>2</td>
<td>8.71</td>
<td>2.81</td>
</tr>
<tr>
<td>Percent Imperviousness</td>
<td>18.8</td>
<td>1.9</td>
<td>10.62</td>
<td>3.7</td>
</tr>
<tr>
<td>Impervious area depression storage (in.)</td>
<td>0.08</td>
<td>0.04</td>
<td>0.07</td>
<td>0.01</td>
</tr>
<tr>
<td>Pervious area depression storage (in.)</td>
<td>0.18</td>
<td>0.08</td>
<td>0.16</td>
<td>0.02</td>
</tr>
<tr>
<td>Soil suction head (in.)</td>
<td>9.7</td>
<td>4.3</td>
<td>7.55</td>
<td>1.42</td>
</tr>
<tr>
<td>Saturated hydraulic conductivity (in./h)</td>
<td>0.32</td>
<td>0.02</td>
<td>0.13</td>
<td>0.09</td>
</tr>
<tr>
<td>Initial moisture deficit (fraction)</td>
<td>0.33</td>
<td>0.3</td>
<td>0.32</td>
<td>0.01</td>
</tr>
<tr>
<td>Groundwater flow coefficient</td>
<td>6x10⁻⁴</td>
<td>5.8x10⁻⁵</td>
<td>1.67x10⁻⁴</td>
<td>1.3x10⁻⁴</td>
</tr>
</tbody>
</table>

To estimate the saturated hydraulic conductivity for subcatchments in the Faneuil Brook sub-basin, soil data was downloaded from the SSURGO database (USDA 2008). A greater part of Faneuil Brook (in the northern side) falls in the Norfolk-Suffolk County, while a smaller part (in the southern side) falls in the Middlesex County. SSURGO data was downloaded from Soil Data Mart for both these counties, which are referred as soil survey areas. Peragallo (1989) gives a brief description of each soil type surveyed for the Norfolk and Suffolk counties. Soil data for each survey area thus downloaded has two subdirectories: tabular and spatial. The subdirectory “tabular” contains a set of ASCII field and text delimited files. Each of these files corresponds to a table in a SSURGO template database, which is documented in the “SSURGO Metadata – Tables” report (USDA 2008). The subdirectory “spatial” contains two feature classes in
the form of ESRI shapefile: Soil Survey Area Boundary Polygons and Map Unit Boundary Polygons for each survey area.

Map units are typically composed of one or more named soil types, different land types and areas of water. Different percent compositions of these entities constitute soil components. Further, soil components are composed of multiple horizons or layers. Therefore, an aggregation method should be adopted if a particular soil attribute that appears in the soil horizon table needs to be represented for map unit polygons in a survey area. Horizon attributes must be aggregated to the component level, before components are aggregated to the map unit level. Also, horizon attributes could be aggregated for the entire soil profile or for a specific depth range (e.g. only surface layer).

The SSURGO data downloaded for the Norfolk-Suffolk and Middlesex counties was processed using the Soil Data Viewer, version 5.1 (USDA 2006). Soil Data Viewer is a tool built as an extension to ArcMap that allows a user to create soil-based thematic maps. In the soil data downloaded from the SSURGO database, the saturated hydraulic conductivity \( K_s \) of different soil components at different horizons is given in the table “chorizon.dbf”. Each soil survey polygon area in the Faneuil Brook sub-basin was identified by a unique map unit key, which consisted of several soil components that were each identified by a unique component key. Further, for each component, there were different values of \( K_s \) at different horizons. Therefore, if a single \( K_s \) value is to be assigned for each map unit, \( K_s \) values first need to be aggregated over a depth range for each soil component, and then the \( K_s \) values for each soil component in a map unit need to be aggregated. This sequential aggregation from the horizon level to the map unit level was executed using the Soil Data Viewer. \( K_s \) values were aggregated over a depth (horizon depth to top) range of 62 to 300 cm. These \( K_s \) values for each component in a
map unit were aggregated using the “Weighted Average” aggregation method in Soil Data Viewer, which resulted in a single $K_s$ value for soil survey polygon area of the Faneuil Brook sub-basin. This vector data was then converted to raster format as before (using the “Feature to Raster” tool), and could be used in ANGel to determine the $K_s$ values for subcatchments at different resolutions. Figure 3.18a presents the raster grids obtained from soil survey polygons in the SSURGO data, while Figure 3.18b shows the raster dataset for $K_s$ overlaid on an artificial network generated by ANGel. Using the Zonal Statistics function in ANGel, each of these subcatchments could be assigned their respective values $K_s$ values.

### 3.8 Import of ANGel Output to SWMM

One of the key advantages of the present version of the ANGel program is that the output is in the form of shapefiles for drainage areas, pipes and drainage nodes. These ESRI shapefiles (.shp format) have all the hydrologic/hydraulic properties required to run a SWMM simulation. The model used in this study was PCSWMM 2010, which is a proprietary interface to SWMM developed by Computational Hydraulics International (CHI). PCSWMM 2010 includes a GIS engine that can work with a number of GIS data formats. The model for an area can be built by using the “Import GIS/CAD” option to import GIS shapefiles for subcatchments, conduits, junctions, outfall, etc and matching corresponding attributes in the project and in the source GIS shapefile. Therefore, the output shapefiles for drainage areas, pipes and nodes that are generated by ANGel for a particular catchment can be easily imported to PCSWMM 2010. Table 3.3 lists the attributes of the output areas, pipes and nodes shapefiles generated by
ANGel that were matched with the corresponding SWMM input parameter using the “Import GIS/CAD” option in PCSWMM 2010.

Table 3.3. List of SWMM input parameter for subcatchments, conduits and junctions that were matched with attributes from the corresponding ANGel output shapefiles.

<table>
<thead>
<tr>
<th>Hydrologic Entity</th>
<th>PCSWMM Project Attribute</th>
<th>Output Attribute from ANGel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subcatchments</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>ID</td>
<td></td>
</tr>
<tr>
<td>Outlet</td>
<td>Outlet</td>
<td></td>
</tr>
<tr>
<td>Area</td>
<td>Area_ac</td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>Width_ft</td>
<td></td>
</tr>
<tr>
<td>Slope</td>
<td>WSLOPE</td>
<td></td>
</tr>
<tr>
<td>Imperv</td>
<td>IMPERV</td>
<td></td>
</tr>
<tr>
<td>Dstore_Imperv</td>
<td>WSTORE1</td>
<td></td>
</tr>
<tr>
<td>Dstore_Perv</td>
<td>WSTORE2</td>
<td></td>
</tr>
<tr>
<td>Suction Head</td>
<td>INFIL1</td>
<td></td>
</tr>
<tr>
<td>Conductivity</td>
<td>INFIL2</td>
<td></td>
</tr>
<tr>
<td>Initial Deficit</td>
<td>INFIL3</td>
<td></td>
</tr>
<tr>
<td>Receiving Node</td>
<td>Outlet</td>
<td></td>
</tr>
<tr>
<td>Groundwater Flow Coefficient</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td><strong>Nodes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Invert Elev.</td>
<td>Inv_elev</td>
<td></td>
</tr>
<tr>
<td><strong>Conduits</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>ID</td>
<td></td>
</tr>
<tr>
<td>Inlet Node</td>
<td>In_Node</td>
<td></td>
</tr>
<tr>
<td>Outlet Node</td>
<td>Out_Node</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Length_ft</td>
<td></td>
</tr>
</tbody>
</table>
In conclusion, it may be summarized that the present version of ANGel is capable of analyzing actual (or existing) drainage networks with their associated drainage areas, and also capable of generating artificial networks at different resolutions. Furthermore, this version is able to determine hydrologic properties of subcatchments and hydraulic properties of nodes and pipes based on certain user input. The nodes, lines and areas shapefiles generated as output from ANGel can be imported to SWMM and hydrologic simulations can be performed at different resolutions using both actual and artificial networks.
3.9 References


Chapter 4. Results of Scale Effects

The models developed at various resolutions based on the existing storm drain network were named “Actual”, and the networks generated at different resolutions using the ANGel program were named “Artificial”. The actual and artificial networks are at three levels of spatial resolution (L1, L2, L3). The original model is referred to as “Original”. The modified version is called “Actual L2”, and lower and higher versions are called “Actual L1” and “Actual L3”, respectively. Similarly, the artificial network that is comparable to Actual L2 is called “Artificial L2”, and the lower and higher versions are called “Artificial L1” and “Artificial L3”, respectively. The GIS representations of the actual and artificial networks at three different resolutions are presented in Figures 4.1a through 4.1f. Table 4.1 summarizes the models with respect to (i) hydrologic properties (e.g. bifurcation ratio), and (ii) model input parameters (e.g. number of subcatchments. The artificial networks at each resolution were generated to approximately match the number of subcatchments and the drainage density of the actual network at the corresponding resolution. The models were run for water years 2000, based on the original study (Zarriello and Barlow 2002). Also, to confirm the results observed in the storms for water year 2000, the models were run for water years 1999 and 2001.
Table 4.1: Differences in hydrologic properties, input parameters and model output between actual and artificial networks at three spatial resolution levels.

<table>
<thead>
<tr>
<th>Model</th>
<th>Level</th>
<th># Sub</th>
<th>$R_b$</th>
<th>$R_r$</th>
<th>DD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phil's Original</td>
<td>2</td>
<td>18</td>
<td>2.75</td>
<td>1.47</td>
<td>1.91E-03</td>
</tr>
<tr>
<td>Actual L1</td>
<td>1</td>
<td>4</td>
<td>2.75</td>
<td>0.78</td>
<td>6.39E-04</td>
</tr>
<tr>
<td>Actual L2</td>
<td>2</td>
<td>19</td>
<td>2.75</td>
<td>1.47</td>
<td>1.91E-03</td>
</tr>
<tr>
<td>Actual L3</td>
<td>3</td>
<td>401</td>
<td>5.2</td>
<td>2.55</td>
<td>9.39E-03</td>
</tr>
<tr>
<td>Artificial L1</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2.08</td>
<td>7.45E-04</td>
</tr>
<tr>
<td>Artificial L2</td>
<td>2</td>
<td>21</td>
<td>3.33</td>
<td>2.07</td>
<td>1.63E-03</td>
</tr>
<tr>
<td>Artificial L3</td>
<td>3</td>
<td>616</td>
<td>4.06</td>
<td>2.23</td>
<td>8.28E-03</td>
</tr>
</tbody>
</table>

#Sub = number of subcatchments; $R_b$ = bifurcation ratio; $R_r$ = length-order ratio; DD = drainage density

Figure 4.1. GIS representations of actual and artificial networks (pipes and nodes) with their corresponding drainage areas. (a) Actual L1 with 4 areas; (b) Actual L2 with 19 areas; (c) Actual L3 with 401 areas; (d) Artificial L1 with 5 areas; (e) Artificial L2 with 21 areas; and (f) Artificial L3 with 616 areas.
Before comparing the output from the different model resolutions, the storm of December 6, 2000, which was used to compare the original model results with data (Zarriello and Barlow 2002) is used to benchmark the Actual L2 model (Figure 4.2). The modified model predicts a higher peak flow, which can be attributed to the removal of the pond. However, the changes introduced for the scaling analysis (discussed above) do not alter the model behavior significantly. The focus of this study was on understanding the effects of spatial resolution rather than on how accurately a particular model simulates the observed data.

There was very little difference (<2%) in the total annual outflow volumes predicted by the different resolutions using both actual and artificial networks. However, significant differences were observed for peak flows, and the subsequent discussion will focus on that.
4.1 Peak Flows for All Storms

4.1.1 Storms in Water Year 2000

The models using actual and artificial networks at different resolutions were run for water year 2000 and peak flows were compared for 50 different storm events. For these simulations the 15-min rainfall data used were obtained from the Boston Water and Sewer Commission (BWSC) precipitation gage at Allston-Brighton in Boston. The peak flows are normalized to those of the highest spatial resolution, L3, for both actual and artificial networks. These results are presented in Figures 4.3a and 4.3b, where the percent difference in peak flows (with respect to L3) for L1 and L2 are plotted against the total rainfall depth.

Figure 4.3. Percent difference in peak flows with respect to (a) Actual L3 and (b) Artificial L3 plotted against total rainfall depth for 50 storms of water year 2000.
Figure 4.3 shows that for the larger storms (in terms of total rainfall depth), the peak flows predicted by the lower resolution models were lower for both actual and artificial networks. For four of the largest storms, the peak flows predicted by Actual L1 and Actual L2 were, respectively 10-28% and 2-9% lower than Actual L3. Using artificial networks, the peak flow predictions by Artificial L1 and Artificial L2 were, respectively 11-26% and 4-16% lower than Artificial L3. As an example, for the storm on July 27 (pair of symbols corresponding to 2.1 in. in Figure 4.3), the peak flows predicted by Actual L1 and Actual L2 were, respectively 28% and 8% lower than Actual L3 (Figure 4.3a). For the same storm, the peak flows predicted by Artificial L1 and Artificial L2 were, respectively 26% and 16% lower than Artificial L3 (Figure 4.3b).

For the smaller storms, the peak flows predicted by the lowest resolution model (L1) were higher for both actual and artificial networks (Figures 4.3a and 4.3b). For these storms, the peak flows predicted by Actual L1 were 1-26% higher than Actual L3. Using artificial networks, the peak flows were 1-14% higher than Artificial L3. However, there were no significant differences between L2 and L3 resolutions for either actual or artificial networks. As an example, for the storm on July 22 (pair of symbols corresponding to 0.32 in. in Figure 4.3), the peak flow predicted by Actual L1 was 26% higher than Actual L3 (Figure 4.3a), while the peak flow predicted by Artificial L1 was 14% higher than Artificial L3 (Figure 4.3b). Both Actual L2 and Artificial L2 predicted peak flows that were very close (< 4%) to their corresponding highest resolutions models. For the smaller storms, in general, it was observed that the difference in peak flows between the coarsest and finest resolutions were lower for artificial networks compared to actual networks.
4.1.2 Storms in Water Years 1999 and 2001

To confirm the results observed in the storms for water year 2000, the models were run for water years 1999 and 2001. For these simulations the 15-min rainfall data used were obtained from the precipitation gage at Logan Airport in Boston. The peak flows were compared for 20 different storms events for each of these years and the results are presented in Figures 4.4a through 4.4d. The peak flows were normalized as before to those of the highest resolution networks.

Figure 4.4. Percent difference in peak flows with respect to (a) Actual L3 and (b) Artificial L3 plotted against total rainfall depth for 20 storms of water year 2000. Percent difference in peak flows with respect to (c) Actual L3 and (d) Artificial L3 plotted against total rainfall depth for 20 storms of water year 2001.
Figure 4.4 shows that for most of the large storms (total rainfall depth > 1.5 in.) in both the water years, the peak flows predicted by the lower resolution models were lower than those predicted by the highest resolution model. For water year 1999, this difference in peak flows for large storms with respect to Actual L3 varied from 8 to 29% for Actual L1 and from 3 to 14% for Actual L2. The difference with respect to Artificial L3 varied from 7 to 30% for Artificial L1 and from 3 to 18% for Artificial L2. Similarly for water year 2001, the peak flows predicted by the lower resolution models for large storms were lower than Actual L3 by 11 to 30% for Actual L1 and by 3 to 12% for Actual L2. The peak flows predicted by Artificial L1 were lower by 10 to 33%, and those predicted by Artificial L2 were lower by 4 to 15% compared to Artificial L3.

On the other hand, for the majority of the smaller storms (total rainfall depth < 1.5 in.), the peak flows predicted by the lower resolution models were higher than those predicted by the highest resolution model. Among the smaller storms analyzed for water year 1999, the peak flows with respect to Actual L3 were higher by 2 to 18% for Actual L1, whereas the peak flows with respect to Artificial L3 were higher by 3 to 10% for Artificial L1. As observed for water year 2000, for small storms the peak flows predicted by resolutions L2 and L3 were not very different (< 5% for actual networks and < 3% for artificial networks). Similarly, for the smaller storms of water year 2001, the peak flows with respect to Actual L3 were higher by 3 to 20% for Actual L1, and by 1 to 11% for Artificial L1 with respect to Artificial L3. Again, as before, the peak flows for small storms predicted by resolutions L2 and L3 were not very different (< 2% for actual networks and < 3% for artificial networks). These simulations confirm that there is a dual scale effect for the different storm events in all three water years 1999, 2000 and 2001.
4.2 Scale Effect in Peak Flows for Large Storms

4.2.1 Large Storms in Water Year 2000

The hyetographs and predicted outflow hydrographs (after pipe routing) and runoff hydrographs (before pipe routing) for a large storm on July 27, 2000 using actual networks at the three spatial resolutions are shown in Figures 4.5a and 4.5b, respectively. The July 27, 2000 storm is a relatively large storm with a total depth of 2.1 in. that builds up gradually, has a total duration of 30 h, and a maximum intensity of 0.48 in./h. For this storm, the lower resolution networks Actual L1 and Actual L2 predict lower peak flows (59 cfs and 76 cfs, respectively) than the highest resolution network Actual L3 (83 cfs). The scale effect with respect to the peak surface runoffs was similar to the peak outflows.

Figure 4.5. (a) Outflow hydrograph and hyetograph, (b) runoff hydrograph for July 27, 2000 storm compared at resolutions Actual L1, Actual L2, Actual L3.
Results of another large storm on June 6, 2000 have been presented in Figure 4.6 to confirm the previous observations. Figures 4.6a and 4.6b show the outflow hydrograph and hyetograph and the surface runoff hydrograph, respectively for the June 6, 2000 storm, compared at model resolutions Actual L1, Actual L2 and Actual L3. The total precipitation depth for the storm on June 6 was 4.4 in which occurred over a duration of 26 h and had a maximum intensity of 0.56 in./h. For this storm, the peak flow predicted by Actual L1 (190 cfs) was 21% lower than Actual L3 (240 cfs), while there was a much smaller difference (approximately 3%) between the peak flows predicted by Actual L2 (232 cfs) and Actual L3 (Figure 4.6a). The scale effect with respect to the peak surface runoffs was similar to the peak outflows. The peak surface runoff predicted by Actual L1 (191 cfs) was 20% lower than Actual L3 (240 cfs), while the difference in peak runoffs between Actual L2 (235 cfs) and Actual L3 was approximately 2% (Figure 4.6b).

Figure 4.6. (a) Outflow hydrograph and hyetograph, (b) runoff hydrograph for June 6, 2000 storm compared at resolutions Actual L1, Actual L2, Actual L3.
4.2.2 Large Storms in Water Years 1999 and 2001

The outflow hydrograph (after conduit routing) at the outlet of the catchment, the rainfall hyetograph and time series of losses (e.g., infiltration, evaporation, evapotranspiration, etc.) and the surface runoff hydrographs (before conduit routing) are shown in Figures 4.7 and 4.8, respectively for two large storms on October 8, 1998 and March 30, 2001. Not only were these storms large in terms of total rainfall depth, but also they built up in a gradual manner. The storm on October 8 was the second largest storm of water year 1999 with a total rainfall depth of 4.35 in., maximum intensity of 0.76 in./h, and total duration of over 18 h. The storm on March 30 was the third largest storm of water year 2001 with a total rainfall depth of 2.18 in., maximum intensity of 0.48 in./h and total duration of over 80 h. The peak outflows predicted by Actual L1 and Actual L2 were, respectively 29% and 7% lower than Actual L3 for the storm on October 8, 1998 (Figure 4.7a). For the storm on March 30, 2001, the peak outflows predicted by Actual L1 and Actual L2 were, respectively 30% and 12% lower than Actual L3 (Figure 4.8a). It was confirmed that for both these large storms, the losses predicted by the lowest resolution model were higher than those predicted by the higher resolution models and this difference was greatest when the rainfall intensity peaked (Figures 4.7b and 4.8b). Consequently, the surface runoff hydrographs at the three resolutions followed the same trend as the outflow hydrograph for both the large storms (Figures 4.7c and 4.8c).
Figure 4.7 (a) Outflow hydrograph; (b) system losses; (c) surface runoff hydrograph for storm on October 8, 1998.

Figure 4.8. (a) Outflow hydrograph; (b) system losses; (c) surface runoff hydrograph for storm on March 30, 2001.
### 4.2.3 Effect of Antecedent Moisture in Large Storms

For the larger storms, for all three water years, another important observation was made. It was observed from Figures 4.3 and 4.4, that some of the large storms did not show the same scale effect as other large storms with comparable rainfall depths. For example, it was noted from Figures 4.3a and 4.3b, that there was one storm with a rainfall depth of 1.56 in. (March 11, 2000), which showed a scale effect opposite to what was observed for another storm with a rainfall depth of 1.51 in. (September 15, 2000).

For the storm on March 11, the peak flows predicted by Actual L1 and Actual L2 were, respectively 7% and 2% higher than the peak flow predicted by Actual L3. The results were consistent using artificial networks. However, for a similar storm (in terms of rainfall depth) on September 15, the peak flows predicted by Actual L1 and Actual L2 were, respectively 15% and 9% lower than Actual L3. The results were again consistent using artificial networks. Similar discrepancies were observed for some large storms in other water years.

This discrepancy in observed scale effect among the large storms could be explained by considering the effect of antecedent moisture conditions. Among the four storms in water year 2000 (Figure 4.3), for which the lower resolution predicted a lower peak flow, all were preceded by shorter dry periods, as compared to the storm on March 11. For the storm on March 11, approximately 288 h had passed since a storm event greater than 0.1 in. depth. The effect of antecedent moisture conditions on the scale effect in large storms was illustrated by plotting the percent difference in peak flows with respect to the highest resolutions (Actual L3 or Artificial L3) as a function of both total rainfall depth and antecedent moisture conditions for the 14 largest storms in water year 2000 (Figure 4.9). The antecedent moisture condition for each of these storms in this figure has been represented as the time elapsed (h) since the last storm of rainfall depth greater than 0.1 in.
It is seen from the above figure that there is a good correlation between the rainfall depth and antecedent moisture conditions for the four storms on the left side of the ‘zero-scale effect’ line (where peak flows predicted by Actual L1 were lower than Actual L3). These were the larger storms on April 21, June 6, July 27 and September 15 in water year 2000. Not only did these storms have high rainfall depths (red squares in Figure 4.9), but they were also all preceded by storms (> 0.1 in.) that occurred within 100 h (approximately 4 days) prior to them. The storms on the right side of the zero-scale effect line (where peak flows predicted by Actual L1 were higher than Actual L3), were in general all preceded by storms (> 0.1 in.) that occurred quite a long time before them. For the storm on March 11, the black dot (288 h) coincided with the red square (1.56 in.) in Figure 4.9. However, between the two effects of rainfall depth and antecedent moisture conditions, the rainfall depth is more important to determine what kind of scale effect will be manifested for a particular storm event. For example, in Figure 4.9, we see there is one storm on May 24 which is preceded by the shortest number of hours (20 h)
since the last storm (> 0.1 in.) among 14 storms analyzed. However, for this storm, the peak flow predicted by Actual L1 was 10% higher than Actual L3. This can be explained because the total rainfall depth for this storm (0.81 in.) was much lower than that of the other larger storms.

To further establish the effect of antecedent moisture on the large storms, Figure 4.3a was re-plotted by removing the effect of antecedent moisture in two large storms on April 22 (3.4 in.) and July 27 (2.1 in.). The results are presented in Figure 4.10. The peak flows predicted by Actual L1 were 10% and 28% lower than Actual L3, respectively for storms on April 22 and July 27. Now, these storms were run with dry antecedent conditions characterized by using values for unsaturated zone moisture content, initial moisture deficit and initial groundwater table equal to those at the start of simulation assuming dry antecedent conditions. In this case the peak flows predicted by the coarsest resolution model (Actual L1) were 14% and 1% higher, respectively for the storms on April 22 and July 27 (shown by the hollow markers in Figure 4.10). Therefore, it would be reasonable to attribute the scale effect observed in large storms to the combined effects of rainfall depth and the moisture conditions prior to that storm event.

Figure 4.10. Percent difference in peak flows predicted by Actual L1 with respect to Actual L3 plotted against total rainfall depth. Peak flows predicted by Actual L1 for storms on April 22 and July 27 run with dry antecedent conditions are shown as hollow markers.
4.3 Scale Effect in Peak Flows for Small Storms

4.3.1 Small Storms in Water Year 2000

The hyetographs and predicted outflow hydrographs (after pipe routing) and runoff hydrographs (before pipe routing) for a small storm on July 22, 2000 using actual networks at the three spatial resolutions are shown in Figures 4.11a and 4.11b, respectively. The July 22 storm is smaller storm with a total depth of 0.32 in. that is more flashy with a total duration of 7 h and a maximum intensity of 0.68 in./h. For this storm, the lowest resolution network Actual L1 predicts a higher peak flow (46 cfs) than the highest resolution network Actual L3 (36 cfs). There was no significant difference in the peak flows between Actual L2 and Actual L3 for this storm. Also, for this storm there was no significant difference in peak runoffs among the three resolutions (Figure 4.11b).

![Outflow hydrograph and hyetograph, runoff hydrograph for July 22, 2000 storm compared at resolutions Actual L1, Actual L2, Actual L3.](image-url)
Results of another small storm on June 2, 2000 have been presented in Figure 4.12 to confirm the previous observations. Figures 4.12a and 4.12b show the outflow hydrograph and hyetograph and the surface runoff hydrograph, respectively for the June 2, 2000 storm, compared at model resolutions Actual L1, Actual L2 and Actual L3. This storm was a classic example of a small storm but flashy in nature. It had a total precipitation depth of 0.56 in which occurred in a duration of 4 h and had a maximum intensity of 1.08 in./h. For this storm, the peak flow predicted by Actual L1 (74 cfs) was 21% higher than Actual L3 (61 cfs), while there was a much smaller difference (approximately 2%) between the peak flows predicted by Actual L2 (60 cfs) and Actual L3 (Figure 4.12a). Unlike the large storms, the scale effect in small storms between the coarsest and the finest resolutions was much lower or almost non-existent in the peak runoff predictions. The peak runoff predicted by Actual L1 (95 cfs) was 6% higher than Actual L3 (90 cfs), and that by Actual L2 (90 cfs) was equal to Actual L3 (Figure 4.12b).

Figure 4.12. (a) Outflow hydrograph and hyetograph, (b) runoff hydrograph for June 2 storm compared at resolutions Actual L1, Actual L2, Actual L3.
4.3.2 Small Storms in Water Years 1999 and 2001

A similar analysis was done for the smaller storms of both the water years 1999 and 2001 and the results are presented for two small storms on July 6, 1999 and September 13, 2001 in Figures 4.13 and 4.14, respectively. Again, as observed in the previous chapter, not only were these storms small in terms of total rainfall depth, but they were also flashy in nature. The storm on July 6 had a total rainfall depth of 0.29 in., maximum intensity of 0.44 in./h, and total duration of approximately 7 h. The storm on September 13 had a depth of 0.51 in., maximum intensity of 0.4 in./h and total duration of approximately 15 h. The peak outflows predicted by Actual L1 were 17% and 20% higher than Actual L3, respectively for the storms on July 6 (Figure 4.13a) and September 13 (Figure 4.14a). However, for both these small storms, there was very little difference between the peak flows predicted by Actual L2 and Actual L3. Unlike the large storms, there were no significant differences in losses among the different resolutions (Figures 4.13b and 4.14b). Again, unlike the large storms, while the peak outflows (after conduit routing) between the coarsest and the finest resolutions differed by almost 20%, the peak surface runoffs (before conduit routing) predicted by Actual L1 were higher than Actual L3 by only 4% and 6%, respectively for the storms on July 6 (Figure 4.13c) and September 13 (Figure 4.14c). The peak runoffs by Actual L2 and Actual L3 almost coincided for both these storms. Therefore, it was verified as before that for the smaller storms there was a scale effect between the peak outflows, which was almost non-existent in the peak runoffs. This again confirmed our hypothesis that the scale effect in smaller storms is due to routing.
4.4 References

Chapter 5. Analysis of Scale Effects

This chapter discusses the detailed analysis of the results of spatial resolution effects that were presented in the previous chapter using actual networks at different resolutions.

5.1 Analysis of Scale Effect in Large Storms

To understand why the peak flow predicted by the low-resolution model is lower than the high-resolution model, single event simulations were run at the different resolutions for each of the four large storms of water year 2000. For these single event simulations, some modifications were introduced such that the peak flows predicted by the single event simulations match those from the continuous simulation. The initial groundwater table elevation at the beginning of a storm was set equal to the final groundwater table elevation determined by running a continuous simulation to the beginning of the storm. The upper unsaturated zone moisture content and the spatially distributed values of the initial moisture deficit fraction were modified for each event to account for the antecedent moisture conditions in the soil due to storms preceding each of these large storms. Finally, an individual hot start file was used for each event that contained the water depths at each node of the drainage system at the beginning of each storm. The version of SWMM (5.0.016) used in this study did not contain the final groundwater state for each subcatchment zone. Therefore, the above modifications to account for the antecedent moisture conditions in the soil were introduced in addition to using a respective hot start file for each storm event.
5.1.1 Water Balance Parameters for Large Storms

To understand the scale effect for the large storms, water balance parameters and flow rates are compared for the July 27, 2000 storm at the three spatial resolutions (Figure 5.1). The peak outflow ($Q_{p,\text{out}}$), peak runoff ($Q_{p,\text{run}}$) and total runoff ($Q_{\text{shed}}$) show similar patterns (decrease with aggregation), and total infiltration ($\text{INFIL}$) shows an opposite pattern, which points to infiltration as the source of the scale effect. There is very little difference among the other water balance parameters such as surface evaporation ($E$), evapotranspiration from the upper ($ETU$) and the lower ($ETD$) saturated zones.

![Figure 5.1](image)

Figure 5.1. Water balance parameters and peak flow rates compared for July 27 storm at resolutions Actual L1, Actual L2, Actual L3. $Q_{\text{shed}}$: total runoff (in.); INFIL: infiltration (in.); E: evaporation (in); ETU: upper zone evapotranspiration (in.); ETD: lower zone evapotranspiration (in.); $Q_{p,\text{run}}$: peak runoff (cfs); $Q_{p,\text{out}}$: peak outflow (cfs).

The results of another large storm on June 6, 2000 have been presented in Figure 5.2 to confirm the previous observations. Figure 5.2 shows water balance parameters for the June 6 storm, compared at model resolutions Actual L1, Actual L2 and Actual L3. Comparing the water balance parameters, $Q_{\text{shed}}$, which is the total depth of surface runoff (overland flow, before conduit routing) from the subcatchments shows
the same pattern as the peak outflow and peak runoff. The runoff depths predicted by Actual L1 (1.12 in.) and Actual L2 (1.49 in.) were, respectively 29% and 6% lower than Actual L3 (1.58 in.). As in the July 27 storm, it is also seen from Figure 5.2 that infiltration from the surface to the sub-surface zone (INFIL) constituted the greatest part of the input rainfall. For a total rainfall depth of 4.4 in. for this storm, the infiltration depths were, respectively 3.14 in., 2.76 in. and 2.67 in. for Actual L1, Actual L2 and Actual L3. There was relatively very little or no scale effect among the other water balance parameters such as surface evaporation (E), evapotranspiration from upper (ETU) and lower (ETD) saturated zone. Similar results obtained from the storm on June 6 as in the storm on July 27 further confirm that the scale effect in peak flows for the large storms is related to infiltration and surface runoff.

![Figure 5.2](image-url)

**Figure 5.2.** Water balance parameters and peak flow rates compared for June 6 storm at resolutions Actual L1, Actual L2, Actual L3. $Q_{\text{shed}}$: total runoff (in.); INFIL: infiltration (in.); E: evaporation (in); ETU: upper zone evapotranspiration (in.); ETD: lower zone evapotranspiration (in.); $Q_{\text{p,run}}$: peak runoff (cfs); $Q_{\text{p,out}}$: peak outflow (cfs).
5.1.2 Effects of Spatial Distribution of Parameters

To determine what causes the differences in infiltration depths and runoff depths among the different resolutions for large storms, the storm on July 27 is revisited here. The approach followed was to analyze the infiltration depths, runoff depths, peak surface runoffs (overland flow before routing) and peak outflows (at outlet after conduit routing) at model resolutions Actual L1, Actual L2 and Actual L3 by spatially varying one input parameter at a time, while keeping all other parameters constant (equal to their respective area-weighted average values). For these simulations, same initial conditions (initial groundwater table elevations, same initial moisture content in the upper unsaturated zone and same initial moisture deficit fraction) were used in the three resolutions corresponding to each storm. The subcatchment input parameters varied one at a time were slope $S$, length of overland flow $LOF$, percent imperviousness $IMP$, impervious area depression storage $IDS$, pervious area depression storage $PDS$, soil suction head $SUCT$, saturate hydraulic conductivity $K_s$, initial moisture deficit (as fraction) $IMD$ and groundwater flow coefficient $A1$. The effects of varying each of these parameters on infiltration depths, runoff depths, peak surface runoffs and peak outflows for the storm on July 27 have been presented, respectively in Figures 5.3a through 5.3d. The conditions when all parameters were constant (all Equal) and when all parameters were spatially distributed (all Grids) were shown in the figures for comparison.
Figure 5.3. Comparison of (a) Infiltration depths, (b) surface runoff depths, (c) peak surface runoffs, and (d) peak outflows among model resolutions Actual L1, Actual L2 and Actual L3 for the July 27 storm, as a function of spatial variability of different subcatchment parameters.

It can be observed from the figures that spatial variability of the parameter saturated hydraulic conductivity $K_s$ compared to any other parameter caused the greatest scale effect among the model resolutions with respect to infiltration depths, runoff depths, peak surface runoffs and peak outflows. When only $K_s$ was spatially distributed, the infiltration depths predicted by resolutions L1 (1.74 in.) and L2 (1.69 in.) were, respectively 5% and 2% higher than the highest resolution L3 (1.61 in.), compared to them being 7% and 3% higher when all parameters were spatially distributed. The runoff depths predicted by resolutions L1 (0.24 in.) and L2 (0.28 in.) were, respectively 24% and 11% lower than L3 (0.31 in.) when $K_s$ alone was spatially distributed compared to them being 28% and 11% lower when all parameters were spatially variable. Peak
surface runoffs (before conduit routing) predicted by L1 (51.48 cfs) and L2 (58.5 cfs) were, respectively 19% and 8% lower than L3 (63.63 cfs) when only $K_s$ was spatially distributed compared to them being 30% and 7% lower when all parameters were spatially distributed. Finally, peak outflows (after conduit routing) predicted by resolutions L1 (50.49 cfs) and L2 (54.68 cfs) were, respectively 15% and 7% lower than L3 (59.06 cfs) when $K_s$ alone was spatially distributed compared to them being 28% and 8% lower when all parameters were spatially distributed. Similar analysis using two other large storms (April 22 and September 15, results not shown) and using artificial networks (results not shown) produced similar results. Therefore, it was concluded that the scale effect observed in large storms could be largely attributed to the spatial variability of $K_s$ values among the different model resolutions compared to the variability of any other single parameter. However, it was observed that while spatial variability of $K_s$ alone could account for a major part of the differences observed among model resolutions with respect to infiltration and runoff depths, it could account for approximately half of the percent difference of the lowest resolution model (L1) with respect to the highest resolution (L3) in case of peak runoffs and peak outflows.

### 5.1.3 Effect of Distributed Property $K_s$

To evaluate why infiltration is higher for a low-resolution model, the corresponding equation and input parameters were examined. In this model, the modified Green-Ampt model (Mein-Larson, 1973) is used. The depth of water that infiltrates before the soil is saturated ($F_s$) and ponding and runoff occur, is a function of soil properties, including suction head ($SUCT$), saturated hydraulic conductivity ($K_s$) and initial moisture deficit ($IMD$), and the rainfall intensity ($i$):

$$F_s = \frac{SUCT \cdot IMD}{\left(\frac{t}{K_s}\right) - 1}$$  \hspace{1cm} (1)
For each time step a subbasin falls in one of two general regimes. (a) If the cumulative infiltration, $F$, is less than $F_s$, all rainfall infiltrates (infiltration rate $f = i$) and there is no runoff. (b) If the cumulative infiltration, $F$ exceeds $F_s$, surface ponding and runoff occur. When ponding occurs, water will continue to infiltrate at a rate equal to the infiltration capacity of the soil, $f_p$ ($f = f_p$):

$$f_p = K_s (1 + SUCT \cdot \frac{IMD}{F})$$  \hspace{1cm} (2)

When a uniform $K_s$ was assigned and either $SUCT$ or $IMD$ were spatially varied, there was very little or no difference in infiltration among the model resolutions. Therefore, the difference in $F_s$ among the subcatchments is mostly a function of $K_s$.

Figure 5.4a shows $F$ (for one wet time step = 15 min) and $F_s$ plotted against $K_s$ for the lowest (Actual L1) and the highest (Actual L3) resolution models at a rainfall intensity of 0.4 in./h. For the low-resolution model, no subcatchments satisfy the criterion $F > F_s$ (Figure 5.4a), and no runoff is generated. For the high-resolution model, there were 11 subcatchments that have $F > F_s$. The figure illustrates that there can be significant differences in the areas that generate runoff among the different resolutions.

As the rainfall intensity ($i$) increases, $F_s$ decreases (equation 1) and $F$ increases. This means that more subcatchments in both resolutions will satisfy the criterion $F > F_s$, and produce runoff. At even higher intensities, the scenario can be reversed, where all subcatchments from the low-resolution model generate runoff, and a few of the high-resolution model do not generate runoff (Figure 5.4b). However, $K_s$ does not only affect what areas produce runoff (Equation 1), but also the infiltration rate (Equation 2), and thus the amount of runoff. In this case (Figure 5.4b), the runoff predicted by the high-resolution model is still higher. Due to the spatial heterogeneity of $K_s$, the average $K_s$ of the subcatchments producing runoff in Actual L1 (all subcatchments) is 0.127 in./h, whereas in Actual L3 it is 0.107 in./h. As a result, the average infiltration depth from
Actual L1 (0.128 in.) is higher than Actual L3 (0.107 in.), and the average runoff from Actual L1 (0.122 in.) is lower than Actual L3 (0.143 in.). This occurs even though the total area that produced runoff was greater in Actual L1 than Actual L3.

Figure 5.4. Comparison of ‘F’ and ‘Fs’ at (a) $i = 0.4$ in./h and (b) $i = 1.0$ in./h for different subcatchments against their respective $K_s$ at different spatial resolutions Actual L1 and Actual L3 considering one time step (=15 min).
5.1.4 Effect of Distributed Property Length of Overland Flow (LOF)

From the above discussion it was concluded that the spatial variability of $K_s$ caused differences in infiltration and runoff between the model resolutions. However, this effect could not explain all of the difference in peak flows for the large storms. This is shown in Figure 5.5 for the July 27 storm. The peak flows predicted by Actual L1, Actual L2 and Actual L3 using spatially variable values for all parameters (all Var in Figure 5.5) are significantly different. When average values were used for all parameters (all Eq in Figure 5.5), the peak flows are equal. This shows that the scale effect is related to the spatial variability of the parameters. When $K_s$ alone was spatially varied, keeping all other parameters constant ($K_s$ in Figure 5.5), there is a difference in peak flows. However, this effect only accounts for part of the difference in peak flows (compare all Var to $K_s$ in Figure 5.5). Next, each of the remaining parameters, including subcatchment slope ($S$), pervious area depression storage ($d_p$), percent imperviousness ($IMP$), and length of overland flow (LOF) were varied in conjunction with $K_s$ ($K_s$ & $S$, $K_s$ & $d_p$, $K_s$ & $IMP$ and $K_s$ & LOF in Figure 5.5) keeping all other parameters constant, and peak flows were compared. It is observed that when $K_s$ and LOF were varied together, the difference between model resolutions was greatest (compare all Variable to $K_s$ & LOF in Figure 5.5). It is to be noted that when LOF alone was spatially distributed among the model resolutions there was very little or no difference between them with respect to peak flow (Figure 5.5). This means that there was a sequential scale effect in peak flows when $K_s$ values were spatially distributed, followed by the spatial distribution of LOF values among the different model resolutions. The reason for this sequential scale effect was due to the combined spatial variability of parameters $K_s$ and LOF.
To understand the effect of LOF, the equation used to simulate surface runoff $Q_{shed}$ from subcatchments is examined. $Q_{shed}$ is calculated using Manning’s equation (James et al. 2005):

$$Q_{shed} = W \frac{1.49}{n} (d - dp)^{5/3} S^{1/2}$$

$$= \frac{A}{LOF} \frac{1.49}{n} (d - dp)^{5/3} S^{1/2}$$

(3)

where $Q_{shed}$ is in cfs, $W$ is the subcatchment width parameter in ft, $A$ is the subcatchment area in sq. ft, LOF is the length of overland flow in ft, $n$ is Manning’s roughness coefficient, $d$ is the water depth (precipitation minus losses due to infiltration, evaporation, etc.) in ft, $dp$ is the depth of depression storage in ft, and $S$ is the subcatchment slope in ft/ft. The LOF for a subcatchment can be defined as the length of the flow path from the hydraulically most distant point of the subcatchment to its outlet.

Figure 5.5. Peak flows for July 27 storm compared at resolutions Actual L1, Actual L2, Actual L3 under spatial variabilities of different parameters. $K_s$: saturated hydraulic conductivity; $S$: slope; $dp$: depression storage, IMP: percent imperviousness; LOF: length of overland flow.
This is a more fundamental property of a subcatchment than the width parameter $W$, which has no physical explanation, and can be calculated by dividing subcatchment area by its LOF.

To determine LOF, one or many such flow paths can be conjectured for a subcatchment or a self-consistent method can be used (James et al. 2005). In this study, the following method was used to determine the LOF parameter for the subcatchments at different resolutions. From the model set up by Zariello and Barlow (2002), LOF for each subcatchment was determined. Next, the spatially variable LOFs (at the original model discretization level) were converted into GIS raster grids, which were used by ANGel to calculate (based on the Zonal Statistics function in ArcGIS) the LOF values for all subcatchments at resolutions L1, L2 and L3 using both actual and artificial networks. Then, for each subcatchment, $W$ was calculated by dividing subcatchment area by its respective LOF value.

The relationship between surface runoff and LOF is non-linear (equation 3), which means it can be subject to the scale effect. The effect of spatial variability of LOF on surface runoff is illustrated in Figure 5.6 using the same time step as for the $K_s$ analysis (Figure 5.4) and a rainfall intensity of 1.0 in./h. To illustrate the effect of variability (or aggregation) of LOF, one subcatchment in Actual L1 and the corresponding subcatchments in Actual L3 were considered. To isolate the effect of LOF, a single $K_s$ value (0.06 in/h) was used. The runoff depths predicted by individual subcatchments are shown by symbols, while the area-weighted average runoff depths are shown by the lines. There is an obvious scale effect. Even though the average LOF is the same in both models, the average runoff depth varies. Therefore, the scale effect for the large storms is a combined effect of the spatial distribution of $K_s$ and LOF.
5.1.5 Combined effects of $K_s$ and LOF at different rainfall intensities

It has been explained in the previous sections how the spatial distribution of $K_s$ and LOF values produce different infiltration and runoff depths when the models are aggregated due to the non-linearity in the infiltration and runoff models. However, in the previous discussion the effects of spatial distribution of $K_s$ and LOF were evaluated separately. This means that when runoff depths at different resolutions were plotted as a function of LOFs, the subcatchments were assumed to have the same $K_s$ values. In this section, the combined effects of $K_s$ and LOFs are discussed, and also their effects at different rainfall intensities are compared.

It was already discussed that the parameter $K_s$ affects infiltration in two ways. It determines which areas produce runoff and also the infiltration rate, and thus the amount of runoff from those areas. Figures 5.7 and 5.8 illustrate the combined effects of $K_s$ and LOF variability on infiltration and runoff depths for the lowest (Actual L1) and the highest (Actual L3) resolution models, respectively at rainfall intensities 0.6 and 1.0 in./h.
5.7a shows $F$ (for one wet time step =15 min) and $F_s$ plotted against $K_s$. $F$ is the potential cumulative infiltration volume for this event, which depends on the rainfall intensity. In these figures, the subcatchments that generate runoff are symbolized using open markers, while the area-weighted average infiltration and runoff depths are shown by lines. At 0.6 in./h intensity, for the low-resolution model (Actual L1), only one subcatchment (area = 139 ac.) satisfies the ponding criterion $F>F_s$ (Figure 5.7a), and generates runoff. For the high-resolution model (Actual L3), there are 190 subcatchments (sum of the areas = 433 ac.) that satisfy the criterion $F>F_s$. The infiltration depths (at $i = 0.6$ in./h for one wet time step = 15 min) under ponding conditions ($f$) at different resolutions are plotted as a function of their corresponding $K_s$ values (Figure 5.7b). The $K_s$ of the single subcatchment that produces runoff in Actual L1 is 0.065 in./h, whereas the average $K_s$ of the 190 subcatchments that produce runoff in Actual L3 is 0.062 in./h. As a result, the average infiltration depth from Actual L1 (0.099 in.) is higher than Actual L3 (0.093 in.). Next, the runoff depths (after overland flow routing) for these subcatchments are plotted against their respective $LOF$s and the results are presented in Figure 5.7c. These runoff depths were calculated considering spatially variable $K_s$ values for the subcatchments. Under such conditions, the average runoff from Actual L1 (0.001 in.) is lower than the average runoff from Actual L3 (0.002 in.). Due to the spatial heterogeneity of $K_s$ and $LOF$s, the runoff depth (after overland flow routing) decreases as a result of spatial aggregation.

A similar analysis was done at a higher rainfall intensity of 1.0 in./h and the results are presented in Figure 4.21. At this intensity, all the subcatchments (whole area = 920 ac.) in Actual L1 satisfy the ponding criterion $F>F_s$ (Figure 5.8a) and generate runoff. For Actual L3, there are 344 subcatchments (sum of the areas = 815 ac.) that have $F>F_s$. As before, the infiltration depths (at $i = 1.0$ in./h for one wet time step = 15
(0.107 in./h). As a result, the average infiltration depth from Actual L1 (0.128 in.) is higher than Actual L3 (0.107 in.). Next, as before, the runoff depths (after overland flow routing) for these subcatchments with variable $K_s$ values are plotted against their respective $LOFs$ and the results are presented in Figure 5.8c. Under the combined spatial variability of $K_s$ and $LOF$, the average runoff depth (after overland flow routing) predicted by Actual L1 (0.003 in.) is lower than Actual L3 (0.008 in.).

Comparing the infiltration and runoff depths predicted by the model resolutions at different rainfall intensities, points to an interesting observation. The relative differences between the model resolutions with respect to infiltration depth and runoff depth increase as the rainfall intensity increases. At 0.6 in./h rainfall intensity, the infiltration depth predicted by Actual L1 is 6% higher than Actual L3, whereas, at 1.0 in/h intensity this difference increases to 20% (comparing Figures 5.7b and 5.8b). Similarly, at 0.6 in./h rainfall intensity, the runoff depth (after overland flow routing) predicted by Actual L1 is 50% lower than Actual L3, whereas, at 1.0 in/h intensity this difference increases to 63% (comparing Figures 5.7c and 5.8c). Therefore, there is a distinct increase in scale effect with respect to infiltration and runoff at higher rainfall intensity, and this could explain the stronger scale effect observed for the larger storms.
Figure 5.7. Comparison of (a) $F$ and $F_s$ as a function of $K_s$, (b) actual infiltration depth $f$ as a function of $K_s$, and (c) surface runoff depth as a function of LOF for subcatchments at resolutions Actual L1 and Actual L3 at rainfall intensity $i = 0.6$ in/h and considering one time step (=15 min).

Figure 5.8. Comparison of (a) $F$ and $F_s$ as a function of $K_s$, (b) actual infiltration depth $f$ as a function of $K_s$, and (c) surface runoff depth as a function of LOF for subcatchments at resolutions Actual L1 and Actual L3 at rainfall intensity $i = 1.0$ in/h and considering one time step (=15 min).
5.2 Analysis of Scale Effect in Small Storms

To understand the scale effect for the smaller storms, a similar approach as the large storms was used. Single event simulations were performed at the different model resolutions for two small storms of water year 2000 (July 22 and June 2) and water balance parameters were compared. Again, as for the large storms, initial conditions were adjusted such that the peak flows predicted by these single event simulations matched those from the continuous simulation.

5.2.1 Water Balance Parameters for Small Storms

The water balance parameters, peak runoff and peak outflow for the storm on July 22 were compared at the three spatial resolutions (Figure 5.9). Unlike for the larger storms, there is no difference in the parameters $Q_{\text{shed}}$, INFIL, E, ETU, and ETD. There is a smaller scale effect (6%) in peak runoffs $Q_{p,\text{run}}$ (before pipe routing) and a larger scale effect (26%) in peak outflows $Q_{p,\text{out}}$ (after pipe routing) between Actual L1 and Actual L3. Also, the time of the peak (Figure 4.10a) was significantly delayed in the highest resolution model (Actual L3).

![Figure 5.9. Water balance parameters and peak flow rates compared for July 22, 2000 storm at resolutions Actual L1, Actual L2, Actual L3. $Q_{\text{shed}}$: total runoff (in.); INFIL: infiltration (in.); E: evaporation (in.); ETU: upper zone evapotranspiration (in.); ETD: lower zone evapotranspiration (in.); $Q_{p,\text{run}}$: peak runoff (cfs); $Q_{p,\text{out}}$: peak outflow (cfs).]
The results of another small storm on June 2 in water year 2000 have been presented in Figure 5.10 to validate the previous observations and the water balance parameters have been compared at three different resolutions. The peak runoffs and outflows have been presented again in Figure 5.10 for comparison. Again, unlike the large storms there was no difference in the surface runoff depths ($Q_{\text{shed}}$) and in the infiltration depths ($\text{INFIL}$) predicted by the three resolutions. The surface evaporation ($E$) and evapotranspiration from the upper unsaturated zone ($\text{ETU}$) were same for all three resolutions, while the evapotranspiration depth from the lower saturated zone ($\text{ETD}$) for Actual L1 (0.132 in.) was 6% lower than Actual L2 and Actual L3 (0.14 in.). Similar results obtained from the two small storms on June 2 and July 22 show that the scale effect observed in smaller storms can be attributed to routing. There is a large difference between the coarsest and finest resolution models with respect to peak outflows (after pipe routing), and a relatively smaller difference between them with respect to peak runoffs (before pipe routing).

![Figure 5.10. Water balance parameters and peak flow rates compared for June 2, 2000 storm at resolutions Actual L1, Actual L2, Actual L3. $Q_{\text{shed}}$: total runoff (in.); INFIL: infiltration (in.); $E$: evaporation (in.); $\text{ETU}$: upper zone evapotranspiration (in.); $\text{ETD}$: lower zone evapotranspiration (in.); $Q_{\text{p,run}}$: peak runoff (cfs); $Q_{\text{p,out}}$: peak outflow (cfs).]
In SWMM there are two kinds of routing: overland flow routing to route the surface runoff from the pervious and impervious sub-areas of a subcatchment to an inlet or pipe; and conduit routing that accounts for routing this overland flow through a system of channels/pipes till it reaches the outlet or an outfall point. The effects of overland flow routing and conduit routing for both these small storms are discussed in greater details in the following sections.

5.2.2 Effect of Overland Flow Routing

In SWMM, overland flow routing for subcatchments is simulated by considering them as non-linear reservoirs. Subcatchments are subdivided into three subareas that simulate impervious areas, with and without depression storage, and pervious area (with depression storage) (James et al. 2005). Each of these areas is approximated as non-linear reservoirs to generate surface runoff or overland flow, which is computed by coupling the continuity equation with Manning’s equation. The continuity equation for a subarea can be written as (James et al. 2005):

\[
\frac{dV}{dt} = A \frac{dd}{dt} = A \cdot i^* - Q
\]  

(4)

where \( V = A \cdot d \) = volume of water on the subarea, \( \text{ft}^3 \),
\( d = \text{water depth, ft} \),
\( t = \text{time, sec} \),
\( A = \text{surface area of subcatchment, ft}^2 \),
\( i^* = \text{excess rainfall intensity, ft/s} \),
\( Q = \text{outflow rate, cfs} \).

The outflow \( Q \) is calculated using Manning’s equation from before:

\[
Q = W \cdot \frac{1.49}{n} (d - dp)^{5/3} S^{1/2}
\]

\[
= \frac{A}{LOF} \cdot \frac{1.49}{n} (d - dp)^{5/3} S^{1/2}
\]

(3)
where \( W \) = subcatchment width, ft
\[ n = \text{Manning's roughness coefficient}, \]
\[ d_p = \text{depth of depression storage, ft}, \]
\[ S = \text{subcatchment slope, ft/ft}. \]

\( \text{LOF} = \text{length of overland flow, ft} \)

Equations (1) and (2) can be combined into one non-linear differential equation that produces the non-linear reservoir equation, which can be solved for depth of flow 'd':

\[
\frac{dd}{dt} = i^* - \frac{1.49}{\text{LOF} \cdot n} (d - d_p)^{5/3} S^{1/2}
\]  
(5)

where \( \text{LOF} \) is equal to subcatchment area divided by its width. \( \text{LOF} \) for a subcatchment can be defined as the length of the flow path from the hydraulically most distant point of the subcatchment to its outlet (TR-55 1986).

It can be seen from equation (3) that overland flow from a subcatchment is a function of the subcatchment's \( \text{LOF} \), which can be related to its hydrologic attributes. One or many such flow paths can be conjectured for a subcatchment or a self-consistent method can be used to determine \( \text{LOF} \) (James et al. 2005). This is a more fundamental characteristic of a subcatchment rather than the width \( W \), which has no physical explanation. The latter should be calculated by dividing subcatchment area by its \( \text{LOF} \).

When subcatchments are aggregated and parts of the drainage network are eliminated, system storage decreases and subcatchment runoff feeds instantaneously into inlets, which results in higher runoff peaks. The length of overland flow (LOF) is considered a measure of storage on the subcatchments (James et al. 2005). It was already discussed that in the low-resolution model (Actual L1), \( \text{LOF} \)s were determined by averaging the \( \text{LOF} \)s from the high-resolution model (Actual L3). However, this aggregation procedure does not account for the decrease in storage in the low-resolution model. This may be compensated for by increasing the \( \text{LOF} \)s (storage) for all the subcatchments in the low-
resolution model.

The effects of increasing LOFs in the low-resolution model on the peak runoff and peak outflows have been presented in Figures 5.11 and 5.12 for the July 22 and the June 2 storms, respectively. The low-resolution model was run by increasing the LOFs 1.2, 1.4 and 2 times their original LOF values, and the run conditions are abbreviated, respectively as \textit{L1\_LOFx1.2}, \textit{L1\_LOFx1.4} and \textit{L1\_LOFx2} in Figures 5.11 and 5.12. All other parameters were kept unchanged and the peak outflow and runoff were compared to the higher-resolution models. For the July 22 storm, it was observed that when the LOFs for the low-resolution model were increased 1.2 times their original values, the peak runoff and peak outflow were, respectively 3% and 21% higher (compared to the original 6% and 26%) than Actual L3. When the LOFs were increased 1.4 times their original values, the peak runoffs matched (difference < 0.5%) the higher resolutions (inset in Figure 5.11b), but the peak outflow was still 17% higher than Actual L3 (Figure 5.11a). When the LOFs were further increased to double their original values, the peak runoff predicted by the low-resolution model decreased significantly and was 9% lower than the higher resolutions, while the peak outflow was still 7% higher.
Figure 5.11. Effect of increasing length of overland flow on (a) outflow hydrograph, and (b) runoff hydrograph for July 22 storm.
For the June 2 storm, it was observed that when the LOFs for the low-resolution model were increased 1.4 times their original values, the peak runoff (Figure 5.12b) matched (difference ~1%) the higher resolution models, while the peak outflow (Figure 5.12a) was 11% higher (compared to the original 21%) than Actual L3. When the LOFs were further increased to double their original values, the peak outflow predicted by the low-resolution model matched (difference ~1%) the higher resolution models. However, under this condition, the peak runoff decreased significantly and was 10% lower than Actual L3.

Figure 5.12. Effect of increasing length of overland flow on (a) outflow hydrograph, and (b) runoff hydrograph for June 2 storm.
An increase in LOF implied a decrease in width (since width = area/LOF), and hence a decrease in the overland flow rate (equation 2). Based on an earlier study in the Canadian SWMM report by Proctor and Redfern (1976), this effect could be physically explained by considering LOF as a measure of the difficulty with which surface runoff from subcatchments will drain to the outlet. If LOF was increased, it implied that it takes longer for the water to reach the subcatchment outlet, causes excess water to be built up on the flow plane in response to a rainfall event, results in greater surface storage, and hence, a lower rate of overland flow. Therefore, with such large increases of LOFs, the peak runoffs predicted by the lowest resolution model were no longer representative of the peak runoff predicted by Actual L1. Such an increase in LOF resulted in water spending longer on an infiltrating surface, more infiltration, and hence reduced peak runoff. In reality, this water is spending time on an impervious surface (gutter or pipe), which implied that infiltration may have to be reduced when LOFs are increased to achieve reasonable agreement between lower and higher resolution models with respect to both peak runoffs and outflows. Therefore, the scale effect observed in small storms cannot be explained by overland flow routing alone, and the effect of conduit routing needs to be considered.

5.2.3 Effect of Conduit Routing

For smaller storms, the larger scale effect in peak outflows compared to peak runoffs can be explained by considering that conduit routing plays a stronger role than overland flow routing. The total conduit length and available storage in pipes is much higher in the highest resolution model compared to the lowest one, which leads to peak flow attenuation. The sum of the conduit lengths in Actual L1 was approximately 15 times smaller than the sum of the conduit lengths in Actual L3. Also, Actual L1 had only 4 nodes compared to 403 nodes in Actual L3. To illustrate the effect of conduit routing, a
new pipe was added and the length of another pipe was increased so that the sum of conduit lengths increased by 45% compared to the original Actual L1. Note that spatial aggregation of subcatchments was not modified and other conduit properties such as conduit diameter, slope and Manning’s roughness parameters were not changed from their initial values. The GIS representation of original Actual L1 and the modified Actual L1 with two new pipes are shown in Figures 5.13a and 5.13b, respectively. Model simulations were performed for the low-resolution model using these longer pipes (run conditions abbreviated as LPipes in Figures 5.14 and 5.15), and the outflow and runoff hydrographs were compared for the storms on July 22 and June 2 in Figures 5.14 and 5.15, respectively. It is to be noted that for the run condition LPipes, all other parameters, including LOF values, were kept unchanged for the low-resolution model and equal to their original spatially variable values.

Figure 5.13. (a) Original pipes in Actual L1. (b) A new pipe added and an original pipe length increased in Actual L1.
For the storm on July 22, the peak outflow (Figure 5.14a) predicted by the low-resolution model with longer pipes decreased to 40 cfs (from the original 46 cfs for Actual L1) and this was 10% higher (compared to original 26%) than Actual L3. Similarly, for the June 2 storm, the peak outflow (Figure 5.15a) predicted by the modified low-resolution model decreased to 64 cfs (from the original 74 cfs for Actual L1) and this was 4% higher (compared to original 21%) than Actual L3. This means that if the conduit lengths were further increased, it could account for the entire scale effect in peak outflows between the low and higher resolution models for the smaller storms. However, as expected, increasing the pipe lengths does not produce any change in the peak runoff predicted by the low-resolution model (Figures 5.14b and 5.15b), and obviously conduit routing cannot explain the scale effect observed in the peak runoffs between model resolutions. Therefore, the combined effects of conduit routing and overland flow routing have to be considered to explain the complete scale effect in small storms.
Figure 5.14. Effect of increasing conduit length and combined effect of increasing conduit length and length of overland flow on (a) outflow hydrograph, and (b) runoff hydrograph for July 22 storm.
Figure 5.15. Effect of increasing conduit length and combined effect of increasing conduit length and length of overland flow on (a) outflow hydrograph, and (b) runoff hydrograph for June 2 storm.
5.2.4 Combined Effect of Overland Flow and Conduit Routing

To explain the complete scale effect in peak runoffs and peak outflows for small storms, the combined effect of both overland flow and conduit routing is illustrated here for the two storms on July 22 and June 2. For the low-resolution model, spatially variable LOF values for the subcatchments were successively increased and the network with longer pipes was used. Simulations were performed, peak outflow and peak runoff were compared with respect to the highest resolution model, and the scenarios where the scale effect have been best accounted for are presented in Figures 5.14 and 5.15. For the July 22 storm, it was observed that when the LOFs for the subcatchments in the low-resolution model were increased 1.4 times their original spatially variable values and when the longer pipes network was used (run condition abbreviated as LPipes_LOFx1.4 in Figure 5.14), both the peak outflow (Figure 5.14a) and the peak runoff (Figure 5.14b) predicted by the low-resolution model matched that of Actual L3 within a difference of less than 1%. Similarly, for the June 2 storm, when the LOFs for subcatchments in the low-resolution model were increased 1.3 times their original spatially variable values and when the longer pipes network was used (run condition abbreviated as LPipes_LOFx1.3 in Figure 5.15), the peak outflow (Figure 5.15a) and the peak runoff (Figure 5.15b) predicted by the low-resolution model matched that of Actual L3 within a difference of 1% and 0.5%, respectively. This clearly illustrates that the scale effect in small storms is due to both overland flow routing and conduit routing. The smaller scale effect in peak runoffs is due to overland flow routing, while the larger scale effect in peak outflows is due to conduit routing.

In summary, we can explain the dual scale effect in large and small storms using two different mechanisms. In large storms, the scale effect in peak flows is due to differences in infiltration and runoff due to the spatial distribution of mainly the
parameters saturated hydraulic conductivity (Ks) and length of overland flow (LOF). In small storms, the scale effect in peak flows is due to the combined effects of overland flow and conduit routing. It is to be noted that the mechanisms responsible for scale effects are different due to the nature of the storm characteristics. In large storms the runoff is generated from pervious areas as well, and therefore, the differences in infiltration are important. However, in smaller storms, runoff is mainly generated from impervious areas that are not affected by infiltration. Therefore, the scale effect in smaller storms is due to routing.

5.3 References


Chapter 6. Comparison of Actual and Artificial Networks

Model simulations were performed with actual sewer networks and artificial sewer networks (generated using ANGel) at different resolutions. This chapter provides a comparison of model results in terms of peak flows and scale effect at varying spatial resolutions using actual and artificial networks.

6.1 Comparison of Peak Flows

The differences between annual total outflow volumes for water year 2000 predicted by the actual and the artificial networks were 5%, 7% and 5%, respectively for resolutions L1, L2 and L3. The comparison of the peak flows for all the 50 storm events for this year have been shown in Figure 6.1. The percent difference in peak flows predicted by the actual and the artificial networks at each resolution for each storm was normalized by the peak flow predicted for that storm by the actual network at the corresponding resolution. These differences were plotted against the total rainfall depth for each storm.

Figure 6.1. Percent difference in peak flows predicted by actual and artificial networks at three resolutions L1, L2 and L3 (normalized by the peak flows predicted by the actual networks at each resolution) as a function of total rainfall depth for each storm.
It can be observed from Figure 6.1 that except for some of the smaller storms, where the peak flows predicted by Artificial L2 were higher than Actual L2, the peak flows predicted by the artificial networks compared relatively well with those of the actual networks at all three resolutions. Figures 6.2a, 6.2b and 6.2c show the outflow hydrograph for the storm on April 22 at the three resolutions L1, L2 and L3, respectively. For this storm, the differences in peak flows predicted by the actual and artificial networks at all three resolutions were less than 1.5%. This was a large storm with a total rainfall depth of 3.38 in and a maximum intensity of 0.44 in./h that occurred for a duration of 58 h.

For comparison, the differences in peak flows between actual and artificial networks were averaged for 50 storms in water year 2000 at each resolution. It was observed that for the lowest resolution models, on an average, the peak flows predicted by Artificial L1 were 1.2% lower than Actual L1. The greatest difference between peak flows at the lowest resolution (L1) was for the storm on November 14, in which the peak flow predicted by Artificial L1 was 6.5% lower than Actual L1. For the highest resolution models, on an average, the peak flows predicted by Artificial L3 were 4.2% higher than Actual L3. The greatest difference between peak flows at the highest resolution (L3) was for the storm on July 22, in which the peak flow predicted by Artificial L3 was 11.4% higher than Actual L3. For the intermediate resolution L2, on an average, the peak flows predicted by Artificial L2 were 5.3% higher Actual L2. The greatest difference between peak flows at the intermediate resolution (L2) was for the storm on June 2, in which the peak flow predicted by Artificial L2 was 17% higher than Actual L2. Another interesting observation was made at L2 resolution for the large storm on July 27. For this storm it was seen that the peak flow predicted by Artificial L2 was 10% lower than Actual L1.
Figure 6.2. Outflow hydrographs for the storm on April 21 predicted by actual and artificial networks at resolutions (a) L1, (b) L2, and (c) L3.
Since the difference in peak flows between actual and artificial networks was observed to be the greatest for the intermediate resolution L2, the reason for the differences observed between Actual L2 and Artificial L2 for the storm on June 2 and July 27 will be discussed below. The outflow (after pipe routing) and runoff (before pipe routing) hydrographs predicted by Actual L2 and Artificial L2 for the June 2 storm are presented in Figure 6.3. It can be observed that there was a small difference (2%) in the peak runoffs between the actual and artificial networks for this storm. However, the difference in peak outflow predicted by Artificial L2 was 17% higher than Actual L2. This implies that the difference in peak flow was due to the combined effects of overland routing and conduit routing. The sum of the conduit lengths in Artificial L2 was 14% lower than Actual L2. To illustrate the combined effect of overland flow routing and conduit routing, the spatially variable LOF values for subcatchments in Artificial L2 were increased 1.1 times their original values and the length of some of the pipes in Artificial L2 were also increased such that the difference in sum of the conduit lengths between actual and artificial networks was 7%. This run condition for the modified artificial network was abbreviated as Art_LPipes_LOFx1.1 in Figure 6.3 and the peak outflow (Figure 6.3a) and peak runoff (Figure 6.3b) were compared. Under this condition, both peak outflow and peak runoff predicted by the artificial network decreased, and consequently their differences with respect to the actual network also decreased.
Figure 6.3. (a) Peak runoff and (b) peak outflow compared using Actual L2 and Artificial L2 and combined effect of increasing conduit length and length of overland flow for June 2 storm.
For the large storm on July 27, the peak flows predicted by Actual L2 and Artificial L2 show a different effect compared to the June 2 storm. The outflow hydrographs, runoff hydrographs and water balance parameters for the July 27 storm have been compared, respectively in Figures 6.4a, 6.4b and 6.4c using Actual L2 and Artificial L2. In this case Artificial L2 predicted a peak flow (69 cfs) that was 9.7% lower than Actual L2 (76 cfs). Similarly, the peak runoff by Artificial L2 (72 cfs) was 12.3% lower than Actual L2 (83 cfs). Among the water balance parameters, there were no significant differences between the actual and artificial networks with respect to total depth of surface runoff depth ($Q_{shed}$), infiltration ($INFIL$), surface evaporation ($E$) and evapotranspiration from the upper saturated zone ($ETU$). However, the evapotranspiration depth from the lower saturated zone ($ETD$) and the total groundwater flow ($GWFLW$) were each 5% higher in Artificial L2 compared to Actual L2. This means that when the artificial network was used, more water was lost due to $ETD$, and more water contributed to sub-surface flow ($GWFLW$) than surface runoff, which consequently resulted in lower peak runoff and peak outflow predicted by Artificial L2 compared to Actual L2.
Figure 6.4. (a) Outflow hydrograph, (b) runoff hydrograph, and (c) water balance parameters for July 27 storm compared using Actual L2 and Artificial L2.
6.2 Comparison of Scale Effect

To analyze the differences in scale effect at different resolutions using actual and artificial networks, Figure 4.3 from the Chapter 4 that shows the summary plot of 50 storms events for the water year 2000 was revisited. In this plot, the differences in peak flows predicted by the models at lower resolutions, L1 and L2, (normalized to those of Actual L3 and Artificial L3, respectively for actual and artificial networks) have been plotted against the total rainfall depth for each of 50 different storms.

6.2.1 Larger Storms

It can be observed from Figures 4.3a and 4.3b that for the larger storms, where the lower resolution models predict a lower peak flow than the highest resolution model, the magnitude of scale effect is quite similar between the actual and artificial networks. The peak flows predicted by Actual L1 for the storms on June 6 (4.4 in.), April 22 (3.4 in.), July 27 (2.1 in.) and September 15 (1.5 in) were, respectively 20%, 10%, 28% and 15% lower than Actual L3. The peak flows predicted by Artificial L1 for the same storms (in the same order) were, respectively 22%, 11%, 26% and 17% lower than Artificial L3. For the storms on June 6 and April 22, the difference in peak flows between Actual L2 with respect to Actual L3, and Artificial L2 with respect to Artificial L3 were approximately equal. For the storm on July 27 (Figure 6.5), it was already discussed in the previous section that the peak flow predicted by Artificial L2 was 10% lower than Actual L2. However, for this storm, the peak flows predicted by Actual L3 and Artificial L3 were approximately equal (difference less than 1%). Therefore, the peak flow predicted by Actual L2 was 8% lower than Actual L3, and the peak flow by Artificial L2 was 16% lower than Artificial L3. The reason for this difference in peak flows for this storm using actual and artificial networks at L2 resolution has already been proposed in the previous section. A similar behavior was observed for the September 15 storm (Figure 6.6). For
this storm, the peak flow predicted by Artificial L2 was 6% lower than Actual L2. Again, there was no significant difference (less than 1.1%) in peak flows between Actual L3 and Artificial L3 for this storm. Therefore, while the difference in peak flows between Actual L2 and Actual L3 was 9%, the difference between Artificial L2 and Artificial L3 increased to 15% for this storm.

Figure 6.5. Outflow hydrographs compared at resolutions L1 and L3 using (a) actual networks, and (b) artificial networks for the July 27 storm.

Figure 6.6. Outflow hydrographs compared at resolutions L1 and L3 using (a) actual networks, and (b) artificial networks for the September 15 storm.
6.2.2 Smaller Storms

It can be observed from Figure 4.3 in Chapter 4 that for the smaller storms, the scale effect between the lowest resolution (L1) and the highest resolution (L3) models was, in general, weaker using artificial networks compared to actual networks. This can be illustrated using two storms on July 22 (0.31 in.) and June 2 (0.56 in.). The difference in peak outflows between the lowest (L1) and the highest (L3) resolution models using actual and artificial networks have been presented in the storms on July 22 and June 2 in Figures 6.7 and 6.8, respectively. For the storm on July 22, the peak outflow predicted by Actual L1 was 26% higher than Actual L3 (Figure 6.7a), while the peak outflow predicted by Artificial L1 was 14% higher than Artificial L3 (Figure 6.7b). For the storm on June 2, the peak outflow predicted by Actual L1 was 21% higher than Actual L3 (Figure 6.8a), while the peak outflow predicted by Artificial L1 was 12% higher than Artificial L3 (Figure 6.8b).

Figure 6.7. Outflow hydrographs compared at resolutions L1 and L3 using (a) actual networks, and (b) artificial networks for the July 22 storm.
The peak flows predicted by Actual L1 and Artificial L1 were equal for the July 22 storm, and their difference was less than 2% for the June 2 storm. However, for the July 22 storm, the peak flow predicted by Artificial L3 was 11% higher than Actual L3. Similarly, for the June 2 storm, the peak flow predicted by Artificial L3 was 10% higher than Actual L3. Consequently, the difference in peak flows between Artificial L1 and Artificial L3 was lower than that between Actual L1 and Actual L3. Since the effect of spatial resolution on smaller storms was more significant when actual networks were used compared to artificial networks, this difference in scale effect could be attributed to routing. The sum of the conduit lengths in Artificial L3 was 12% lower than the sum of the conduit lengths in Actual L3. This implied that less water was lost in the pipes due to conduit storage in Artificial L3, which would result in higher peak flows and consequently lower scale effect compared to Actual L3.

In summary, it can be concluded that the peak flow predictions using actual and artificial networks are comparable for most of the storm events at the different resolutions. Also, the scale effect results are consistent using actual and artificial networks. This implies that artificial networks may be an useful tool for spatial scaling analysis.
Chapter 7. Summary and Recommendations for Further Studies

7.1 Summary

Hydrologic models are widely used to simulate or predict the hydrologic response of a watershed to precipitation events. Spatial heterogeneities in watershed characteristics (e.g. slope, percent imperviousness, etc.) are captured in input model parameters by dividing a watershed into several smaller sub-watersheds, which is referred to as model subdivision. The extent to which a model is subdivided determines its level of spatial resolution. It is well-established that spatial resolution affects model results. Urban hydrologic models involve numerous dynamic and interacting non-linear processes and many spatially varying parameters that are subject to the "scale effect". Although previous researches have documented scale effects in urban hydrology, a general consensus on how scale affects model results has not been reached. Also, the underlying mechanism(s) responsible for observed scale effects were typically not explored in past studies. Since hydrologic processes in the urban environment operate over a range of spatial scales, it is important for the modelers to understand the effects of spatial resolution on simulation results.

The main objective of this research was to characterize and understand the effects of spatial resolution on model results for an urban catchment. The study area used in this research was the Faneuil Brook sub-basin of the lower Charles River watershed in Boston. The general approach that was used involved developing hydrologic models at various spatial resolutions (from four to 616 subcatchments) using both actual and artificial (generated using a fractal algorithm) sewer networks, perform continuous model simulations for different water years (1999, 2000 and 2001) using the
Stormwater Management Model (SWMM), and compare results with respect to total annual outflow volume and peak flows for different storm events. Also, the results for peak flows were compared between actual and artificial networks. The underlying mechanism(s) that are responsible for producing the scale effect have been explained in this study, and the major conclusions of this research are discussed here.

- It is re-established in this study that the level of spatial resolution of a catchment (i.e., degree of aggregation of subcatchments with their corresponding pipes and nodes) produces different effects on different model output parameters. It was observed total annual outflow volume predicted by the models was insensitive to spatial resolution using both actual and artificial networks.
- Spatial resolution had a significant effect on the predicted peak flows. However, this effect was not consistent for all storm events and a distinct difference in scale effect was observed based on storm characteristics.
- For the storms that were larger in terms of total rainfall depth, built up gradually and occurred over a longer duration, the peak flows decreased due to spatial aggregation. These peak flows were significantly lower (by up to 28%) for the larger storms in water year 2000 using actual networks. Consistent results were observed for other water years and using artificial networks.
- For storms that were smaller in terms of total rainfall depth, flashy and occurred over a smaller duration, the peak flows increased due to spatial aggregation. These peak flows were significantly higher (by up to 26%) for the smaller storms in water year 2000 using actual networks. Again, the results were consistent for other water years and using artificial networks.
- The scale effect for the larger storms is a combined effect of the spatial distribution of mainly the parameters saturated hydraulic conductivity ($K_s$) and
length of overland flow (LOF). Spatial heterogeneity in \( K_s \) among the different resolutions determines what areas produce runoff and also the infiltration rate. The predicted average infiltration depth increases due to spatial aggregation, which results in lower peak flows at lower resolutions.

- Spatial heterogeneity in LOF affects surface runoff (or overland flow) since the relationship between surface runoff and LOF is non-linear. The predicted average runoff depth decreases due to spatial aggregation, which also results in lower peak flows at lower resolutions. The scale effect in larger storms is stronger as the rainfall intensity increases.

- The scale effect for the smaller storms is due to overland flow and conduit routing. Smaller storms show a smaller scale effect in peak runoff (before pipe routing) and a stronger scale effect in peak outflow (after pipe routing). The average LOFs in a low-resolution model does not account for the decrease in system storage due to subcatchment aggregation and elimination of parts of the drainage network. As a result, overland flow feeds instantaneously into inlets and produces higher peak runoff. The scale effect in peak runoff can be compensated by increasing the LOFs for the low-resolution model, which in turn increases the system storage.

- The scale effect in peak outflow for smaller storms is due to conduit routing and can be attributed to the decrease in conduit storage as a result of fewer pipes in the low-resolution model. Increasing the conduit lengths in the low-resolution model compensates for this storage and attenuates the peak flows.

- The mechanisms responsible for scale effects are different in large and small storms due to their inherent characteristics. In large storms, the runoff is generated from pervious areas as well, and therefore, the differences in infiltration among model resolutions are more important. However, in smaller
storms, runoff is mainly generated from impervious areas that are not affected by infiltration. Therefore, the scale effect in smaller storms is due to routing.

- The peak flows predicted by the actual and artificial networks at different model resolutions were quite comparable for majority of the storms. For the storms analyzed in water year 2000, the peak flows predicted using actual and artificial networks differed, on an average, by 1%, 5% and 4%, respectively for the lowest, intermediate and highest resolution models. Also, the results of the scaling analysis were consistent using artificial networks for all three water years analyzed. This implies that artificial networks could be applied as a useful tool to evaluate the effects of spatial resolution in urban hydrologic simulations.

In summary, it may be concluded that the underlying mechanism(s) that were identified to be responsible for scale effects could be extrapolated to another study area with similar watershed characteristics. Although, the exact results in terms of magnitude of scale effects for different storms may not be the same, it is expected that the pattern of scale effect at different resolutions in response to large and small storm events will be consistent.

### 7.2 Recommendations for Further Studies

The program Artificial Network Generator (ANGel), which was used to analyze actual drainage networks and generate artificial networks, can be improved considerably. In the present version, artificial sewer networks are generated using the Tokunaga fractal algorithm. Other space-filling fractal algorithms such as H-tree and Peano fractals could be implemented.
The study on effects of spatial resolution could be extended to determine a threshold or optimum resolution, beyond which subdivision produces little or no effect on peak flow predictions. Identifying such a threshold level is of particular interest to modelers to address the appropriate level of subdivision. For this threshold analysis, models would have to be developed for a large number of spatial resolutions, hydrologic simulations performed and peak flows analyzed. Such a study could also be a suitable application of artificial sewer networks, where the modeler can easily generate networks at multiple resolutions without requiring the exact configuration of drainage area, pipes and nodes in the network.

The effects of spatial resolution were studied for the Faneuil Brook catchment in the Boston area. Similar analysis could be conducted using actual and artificial drainage networks for other Boston-area catchments (e.g., Laundry Brook, Stony Brook, and Muddy River) and other urban cities in major US climate regions. Using catchments that span over a wide range of scales and other characteristics (e.g., precipitation, slope, storm drain densities, etc.), would be a worthwhile contribution to examine if the effects of spatial resolution on model results (e.g. peak flow) are similar with respect to, say storm characteristics. This would also be good validation of the mechanisms proposed in this study to understand the scale effect.

The study of spatial resolution effects was conducted using the model SWMM. Although similar results are expected from other models, this needs further verification. It would be an important contribution to perform a similar spatial scaling analysis using other urban hydrology models such as MOUSE and Infoworks. This would be valuable for model comparisons often done in rural hydrology, but not so common in urban hydrology.
Appendix A1: ANGel Tutorial

This document contains a simple step-by-step tutorial for running the ANGel artificial network generator. The user should have ArcGIS 9.2 with the 3-D Analyst extension installed and the following files:

- ArcGIS map document file: ANGel.mxd
- TutArea shapefile (.shp, .shx, .sbx, .sbn, .prj, .dbf)
- TutActual shapefile (.shp, .shx, .sbx, .sbn, .prj, .dbf)

The tutorial files are available from the corresponding author.

Completing the tutorial will take about ten minutes. The tutorial consists of the following three parts:

A. Load program and generate default network.
B. Generate high-resolution artificial network.
C. Expand existing network.

**Part A: Load program and generate default network**

**Step 1.** Open the file ANGel.mxd in ArcGIS and load the shapefile “TutArea.shp”, which is the catchment area, and the shapefile “TutActual.shp”, which is the actual sewer network.
Step 2. Click on the button “Run ANGel” to start ANGel. The ANGel User Input Form appears on the screen as shown below.

Step 3. Click OK to accept all default input. The generated line fractal with the default name “LFractal1” is automatically added as a new layer to the Table of Contents on the
map and the lines are also automatically colored by their Strahler stream order. The output is shown below. You may turn off the “TutActual” Layer to view the output better.

![Map with colored lines](image)

**Part B: Generate high resolution artificial network**

**Step 4.** Now we will explore the different options on the form. Click on the button “Run ANGel” again and this time specify the starting network by selecting the “User Specified” option. Then click the “Click It!” button beside Start X/Start Y and when the form hides, click on the map close to the actual outlet (look at the actual network “TutActual.shp”). The form appears again and you can see the “Start X” and “Start Y” text boxes have been populated with the coordinates of the point you clicked on the map. Similarly, click on the “Click It!” button beside End X/End Y and when the form hides, click on the map close to the centroid of the catchment area. The form appears again and this time you can see the “End X” and “End Y” text boxes have been populated by the coordinates of the point you clicked on the map.

**Step 5.** In the Type section, the default is set to “Tokunaga” and the other options are not available in the present version of ANGel.
**Step 6.** In the **Number of Generations to Add** section, enter 3 as number of generations to add to the starting network.

**Step 7.** In the **Advanced Options** section, enter 4 as the **Spatial Tolerance** value. The spatial tolerance is a function of the spatial accuracy of the input shapefiles.

**Step 8.** Select the “Clip to Basin Area” option and select the “TutArea” shapefile from the pull-down menu. This option clips the generated network to the extents of the catchment area. Also, note that the pull-down menu is automatically populated with the area shapefiles you have on the map document, and you can select the catchment area to which you want the fractal to be clipped.

**Step 9.** Select the “Irregular” option to generate an irregular network. Leave the percentage irregularity to the default value of 30. You may increase or decrease the percent irregularity. However, more than 50 percent irregularity may introduce abnormalities in the network. For a new irregular fractal every time, leave the seed value to -1. For the same irregular network enter a seed value other than -1 (e.g., 1, 2, etc.).

**Step 10.** In the **Output section**, select the “Point”, “Line”, “Area” options to add the point, line and area shapefiles, respectively to the map document. If you want to know the statistics of the generated fractal network, check on the “Stats” option.

**Step 11.** In the “Filename” textbox accept the default pathname and filename of the line fractal network.

**Step 12.** Click **OK**. The generated line fractal with the default name “LFractal2” is automatically added as a new layer to the Table of Contents on the map and the lines are also automatically colored by their Strahler stream order. The point shapefile “PFractal2” and the area shapefile “AFractal2” are also added to the map automatically. The output of the generated fractal network with the lines, points and areas and the output of the statistics text file “Fractal2” are shown below.
The output shown below was generated with the seed option set to 1. Note that your output may be slightly different because of a different random number sequence (-1 seed option).
Part C: Expand Existing Network

**Step 13.** In this last part we will explore how ANGel can also be run to generate an artificial network, by expanding an existing network. Turn off all other layers except “TutArea” and “TutActual”. Click on the button “Run ANGel”.

**Step 14.** This time specify the starting network by clicking on the “Existing Shapefile” option. Select that “TutActual” shapefile from the pull-down menu. Repeat step 5 as above.

**Step 15.** In the **Number of Generations to Add** section, enter 4 as number of generations to add to the existing network.

**Step 16.** Repeat steps 7 through 12. Note that for a different shapefile as the starting network, the spatial tolerance may need to be set at a lower value depending on how accurate the shapefile is.

The generated line fractal with the default name “LFractal3” is automatically added as a new layer to the Table of Contents on the map and the lines are also automatically colored by their Strahler stream order. The point shapefile “PFractal3” and the area shapefile “AFractal3” are also added to the map automatically. The output of the generated fractal network with the lines, points and areas and the output of the statistics text file “Fractal3” are shown below.

The output shown below was generated with the seed option set to 3. Note that your output may be slightly different because of a different random number sequence (-1 seed option).
Appendix A2. Computer Code for ANGel
Option Explicit

Dim IrrSeed As Double
Dim MaxN As Double
Dim MaxNP As Double

Public ClickX As Double
Public ClickY As Double
Public nClick As Integer
Public clicks As Integer
Public ClickWhat As Integer

'Initialize variables for default start and end points

Public DefaultStartX As Double
Public DefaultStartY As Double
Public DefaultEndX As Double
Public DefaultEndY As Double

'Initialize variables for fractal generation
Dim dx_1 As Double
Dim dy_1 As Double
Dim dx_2 As Double
Dim dy_2 As Double

Dim iA As Integer
Dim iA2 As Integer
Dim iB As Integer
Dim iC As Integer
Dim nLinesA As Long
Dim nLinesB As Long
Dim nLinesC As Long
Dim nAreas As Long
Dim nOrder As Long
Dim nStr As Long
Dim nGen As Long

Dim I As Integer
Dim j As Integer
Dim k As Integer
Dim fracType As String
Dim Irr As Boolean
Dim IrrPer As Double
Dim jcount As Integer

' --- file name variables ---

'---------------------- FullLPathName
'-------- PathName
'         -------------- LFileName
'         --------- FileNameBase
'c:\temp\LFractal17.shp
Dim PathName As String
Dim FileNameBase As String
Dim FullLPathName As String
Dim LFileName As String
Dim PFileName As String
Dim AFileName As String

Dim LFractalA() As FLine 'used for complete fractal
Dim LFractalB() As FLine 'used for complete fractal
Dim LFractalC() As FLine 'used for generation 1
Dim OldFLine As FLine
Dim NewFLine As FLine
Dim FLine1 As FLine
Dim FLine2 As FLine

Dim PFractalA() As FPoint
Dim NewFPoint As FPoint

Dim SpaceTol As Double
Dim xIdentifyInts As Double
Dim xShortenInts As Double
Dim xRemShortLines As Double
Dim xClipBuffer As Double

Dim StartTime As Date

'setting a user defined variable for fractal
Private Type FLine
  PLine As IPolyline  'set by Expand1Gen, user specified or
  'existing shapefile
  nPoints As Integer
  Generation As Integer  '
  ID As Long  ' set by CalcTopology
  DownID As Long ' -
  NUpstr As Integer ' -
  outlet As Integer  ' set by FindOutlet or ReadLFractal
  Temp As Integer  '
  Orphan As Integer
  Length As Double
  FlowLength As Double
  StrOrder As Double
  StrID As Long
  Area As Double
End Type

Private Type FPoint
  X As Double
  Y As Double
  ID As Long
End Type
Private Sub cmdOK_Click()

' --- initiator & ngen ---
SpaceTol = Val(txtSpaceTol.Value)
xIdentifyInts = SpaceTol
xShortenInts = SpaceTol * 20
If txtUserGen.Value = 0 Then
    xRemShortLines = SpaceTol * 2
Else
    xRemShortLines = SpaceTol * 20
End If
xClipBuffer = SpaceTol * 20

nGen = Val(txtUserGen.Value) + 1

'If the starting network is user specified...
If btnUserSpec.Value Then
    iA = 1
    nLinesA = 1
    MaxN = nLinesA * (1 + (4 / 3) * (4 ^ (nGen - 1) - 1))
    MaxNP = MaxN + 1

    ReDim LFractalA(MaxN)
    ReDim LFractalB(MaxN)
    ReDim LFractalC(MaxN)
    ReDim PFractalA(MaxNP)

    '...Assign the initial attributes to the initiator
    With LFractalA(iA)
        .PLine = New Polyline
        .PLine.FromPoint = CreatePt(Val(txtStartX.Value), Val(txtStartY.Value))
        .PLine.ToPoint = CreatePt(Val(txtEndX.Value), Val(txtEndY.Value))
        .Generation = 1
        .ID = 1
        .DownID = 0
        .NUpstr = 0
        .outlet = 1
    End With
Else

    'Read the shapefile if the initial network is an existing
'shapefile
Dim inFractName As String
inFractName = cboExShapeFile.Value

Call ReadLFractal(inFractName, True)

'Read the shapefile if user chooses to use existing area
'shapefile with the existing network

Dim inAreaFileName As String
inAreaFileName = cboExAShp.Value

'Call ReadAFractal(inAreaFileName, True)

MaxN = nLinesA * (1 + (4 / 3) * (4 ^ (nGen - 1) - 1))
MaxNP = MaxN + 1

ReDim LFractalB(MaxN)
ReDim LFractalC(MaxN)
ReDim PFractalA(MaxNP)

'Make a copy of all lines from LFractalA to LFractalB,
'redimension LFractalA, and copy back the lines from
'LFractalB to LFractalA
For iA = 1 To nLinesA
    LFractalB(iA) = LFractalA(iA)
Next iA
ReDim LFractalA(MaxN)
For iA = 1 To nLinesA
    LFractalA(iA) = LFractalB(iA)
Next iA

Call CalcTopology
Call FindOutlet
Call SnapJunctions
Call CalcLength

'This is done only if the user chooses to combine the pipes in the
'existing network and also if the user chooses to have pipes in the
'network no longer than a specified value
If chkCombine.Value = True Then
    Call DissolveLines
    If chkSplit.Value = True Then
        Call CalcLength
        Call SplitLongLines
    End If
End If
End If

'note: IrrSeed = -1 is new random number everytime
Irr = chkIrr.Value
IrrPer = Val(txtIrrPer.Value)
IrrSeed = Val(txtSeed.Value)
Rnd (-1)
If IrrSeed > 0 Then
  Randomize (IrrSeed)
Else
  Randomize
End If

, ,
, ' --- get file names ---
, ,
FullPathName = txtFileName.Value
LFileName = FunctionGetFileName(FullPathName)
FileNameBase = Mid(LFileName, 2, Len(LFileName) - 5)
PathName = Left(FullPathName, Len(FullPathName) - Len(LFileName))
PFileName = "P" & FileNameBase & ".shp"
AFileName = "A" & FileNameBase & ".shp"

, ,
, '---For each generation, these get done always, even if there
'is no clipping
, ,
Dim NumInts As Long

For k = 2 To nGen

  Call Expand1Gen
  Call CalcTopology

Dim NumIntsPrev As Integer
NumIntsPrev = -9
Do While True
  NumInts = IdentifyInts
  If NumInts = NumIntsPrev Then
    Debug.Print "ANGel: Error: Unable to remove intersections, continuing with them."
    Exit Do
  End If
  NumIntsPrev = NumInts
  If NumInts > 0 Then
    'Call RemoveInts
    Call ShortenInts
Call CalcLength
Call RemoveShortLines

Else
  Exit Do
End If

Loop

Next k

'Call ReadLFractal("L" & FileNameBase & "a.shp", True)

'If the user decides to clip the fractal network to the extents of his/her basin, only then the following are executed:
If chkClip.Value = True Then
  Call WriteLFractal("a", False)
  Call MakeBuffer
  If nGen > 1 Then
    Call ClipLFractal
    Call ReadLFractal("L" & FileNameBase & "b.shp", False)
    Call DeleteClippedlayer
  Else
    Call ReadLFractal("L" & FileNameBase & "a.shp", False)
  End If
  'Call ReadLFractal("L" & FileNameBase & "b.shp", False)
End If

Call CalcLength
Call RemoveShortLines

'Topology gets calculated always, whether there is clipping or not
Call CalcTopology

Call FindOrphans
Call RemoveOrphans

Call CalcLength
Call CalcFlowLength
Call CalcStrOrder
Call CalcStrID
Call FindOutlet

Call MakePFractal

Call WritePFractal(""

Call DeleteOutlet
Call WriteAFractal(""")

Call WriteLFractal("", True)

Call CalcDrainageLineAttributes

Call CalcNodeID

Call CalcNodeInvElev

Call FindOutletforEachSubcatchment

If chkSlp.Value = True Or chkImperv.Value = True Or chkIDS.Value = True Or _
chkPDS.Value = True Or chkSuct.Value = True Or chkHydcon.Value = True Or _
chkSmdmax.Value = True Or chkA1.Value = True Or chkL.Value = True Then
chkPoint.Value = True
chkArea.Value = True
End If

If chkArea.Value = True Then
  Call AddAShapeFile
End If

If chkPoint.Value = True Then
  Call AddPShapeFile
End If

If chkSlp.Value = True Then
  Call ZonalStatistics(cboSlope.Value, "WSLOPE")
End If

If chkImperv.Value = True Then
  Call ZonalStatistics(cboImperv.Value, "IMPERV")
End If

If chkIDS.Value = True Then
  Call ZonalStatistics(cboIDS.Value, "WSTORE1")
End If

If chkPDS.Value = True Then
  Call ZonalStatistics(cboPDS.Value, "WSTORE2")
End If

If chkSuct.Value = True Then
  Call ZonalStatistics(cboSuct.Value, "INFIL1")
End If
If chkHydcon.Value = True Then
    Call ZonalStatistics(cboHydcon.Value, "INFIL2")
End If

If chkSmdmax.Value = True Then
    Call ZonalStatistics(cboSmdmax.Value, "INFIL3")
End If

If chkA1.Value = True Then
    Call ZonalStatistics(cboA1.Value, "A1")
End If

If chkL.Value = True Then
    Call ZonalStatistics(cboL.Value, "LOF")
    Call CalcWidth
End If

If chkLine.Value = True Then
    Call AddLShapeFile
    Call ColorLShapeFile
End If

If chkStats.Value = True Then
    Call CalcStats
End If

If chkSWMM.Value = True Then
    Call WriteSWMM
End If

Me.Hide

End Sub
Public Sub ReadLFractal(readName As String, ReadAll As Boolean)

Debug.Print "ANGel: Entering ReadLFractal."

StartTime = Now()
Dim pWorkspaceFactory As IWorkspaceFactory
Dim pFeatureWorkspace As IFeatureWorkspace
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass

'Specify the workspace and the feature class
Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
Set pFeatureClass = pFeatureWorkspace.OpenFeatureClass(readName)

'Prepare a feature layer
Set pFeatureLayer = New FeatureLayer
Set pFeatureLayer.FeatureClass = pFeatureClass

Dim pFLayer As IFeatureLayer
Set pFLayer = pFeatureLayer

Dim pFClass As IFeatureClass
Set pFClass = pFLayer.FeatureClass

Dim pFCursorA As IFeatureCursor
Set pFCursorA = pFClass.Update(Nothing, True)

Dim pFCursorB As IFeatureCursor
Set pFCursorB = pFClass.Update(Nothing, True)

Dim pFeature As IFeature

Dim pFields As IFields
Set pFields = pFClass.Fields

Dim Generation_Field As Long
Generation_Field = pFClass.FindField("Generation")

Dim Outlet_Field As Long
Outlet_Field = pFClass.FindField("Outlet")

Dim myLine As IPolyline
Dim pGeoms_Polyline As IGeometryCollection
Dim pClone As IClone
Dim pSegs_Path As ISegmentCollection
Dim xPLine As IGeometryCollection

' --- redimension arrays ---
'
' count lines in shape file
',
nLinesA = 0
Set pFeature = pFCursorA.NextFeature
Do Until pFeature Is Nothing
  Set pFeature = pFCursorA.NextFeature
  nLinesA = nLinesA + 1
Loop
'
' add gen 1, if applicable
',
If Not ReadAll Then
  nLinesA = nLinesA + nLinesC
End If
'
' redim arrays
',
ReDim LFractalA(nLinesA)
ReDim LFractalB(nLinesA)
ReDim PFractalA(nLinesA)
'
' --- populate fractal ---
',
If Not ReadAll Then
  For iA = 1 To nLinesC
    LFractalA(iA) = LFractalC(iA)
    Next iA
Else
  iA = 1
End If

Set pFeature = pFCursorB.NextFeature
Do Until pFeature Is Nothing
  With LFractalA(iA)
    Set myLine = pFeature.Value(1)
    Set pClone = myLine
    Set pGeoms_Polyline = pClone.Clone
    Set xPLine = New Polyline
    For j = 0 To pGeoms_Polyline.GeometryCount - 1
Set pSegs_Path = New Path
pSegs_Path.AddSegmentCollection pGeoms_Polyline.Geometry(j)
xPLine.AddGeometry pSegs_Path
Next j
Set .PLine = xPLine

If Not Outlet_Field = -1 Then
    .outlet = pFeature.Value(Outlet_Field)
End If
If ReadAll Then
    .Generation = 1
Else
    .Generation = pFeature.Value(Generation_Field)
End If
Set pFeature = pFCursorB.NextFeature
iA = iA + 1
End With
Loop

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."
End Sub
Public Sub CalcTopology()

Debug.Print "ANGel: Entering CalcTopology."

StartTime = Now()
', ' --- assign id ---
',

For iA = 1 To nLinesA
    LFractalA(iA).ID = iA
    LFractalA(iA).DownID = 0
    LFractalA(iA).NUpstr = 0
Next iA
',

' --- assign downid ---
',

Dim FoundDownLine As Boolean
For iA = 1 To nLinesA
    FoundDownLine = False
    For iA2 = 1 To nLinesA
            LFractalA(iA).DownID = LFractalA(iA2).ID
            LFractalA(iA2).NUpstr = LFractalA(iA2).NUpstr + 1
            FoundDownLine = True
            Exit For
        End If
    Next iA2
Next iA

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Public Sub FindOutlet()
Debug.Print "ANGel: Entering FindOutlet."

StartTime = Now()

I = 0
'For each line in LFractalA, check its downstream ID and count
'only those as outlet which have nothing downstream of them
For iA = 1 To nLinesA
    If LFractalA(iA).DownID = 0 Then
        Debug.Print LFractalA(iA).ID
        LFractalA(iA).outlet = 1
        I = I + 1
    End If
Next iA
If I > 1 Then
    Debug.Print "ANGel/FindOutlet: Error: Multiple outlets found ", I
End If

Debug.Print "Finished at " & Now() & " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Public Sub SnapJunctions()

Debug.Print "ANGel: Entering SnapJunctions."

StartTime = Now()

For iA = 1 To nLinesA
    For iA2 = 1 To nLinesA
        If LFractalA(iA).DownID = LFractalA(iA2).ID Then
            LFractalA(iA).PLine.FromPoint = LFractalA(iA2).PLine.ToPoint
            Exit For
        End If
    Next iA2
Next iA

Debug.Print "Finished at " & Now() & " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."
End Sub
Public Sub CalcLength()

Debug.Print "ANGel: Entering CalcLength."

StartTime = Now()

For iA = 1 To nLinesA

    LFractalA(iA).Length = LFractalA(iA).PLine.Length

Next iA

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

End Sub

Public Sub DissolveLines()

' descr.: this function merges all lines that have 1 upstream line with their upstream line

Debug.Print "ANGel: Entering DissolveLines."

StartTime = Now()

' note: .Temp used to mark lines already dissolved

Dim FoundOneUp As Boolean
Dim iMax As Integer
Dim Pts As IPointCollection
Dim Pts2 As IPointCollection

' mark all as "not already dissolved"
For iA = 1 To nLinesA
    LFractalA(iA).Temp = 0
Next iA

I = 1
iMax = 100 'XXX NEEDS FIX

' loop until there are no more lines with 1 upstream line
Do While True

    FoundOneUp = False

Next iA
For iA = 1 To nLinesA
    If LFractalA(iA).Temp = 0 Then
        If LFractalA(iA).NUpstr = 1 Then 'found a line with one upstream line
            FoundOneUp = True
            For iA2 = 1 To nLinesA
                If LFractalA(iA2).Temp = 0 Then
                    If LFractalA(iA).ID = LFractalA(iA2).DownID Then 'found the line upstream of that
                        LFractalA(iA2).Temp = 1
                        Set Pts = LFractalA(iA).PLine
                        Set Pts2 = LFractalA(iA2).PLine
                        Pts2.RemovePoints 0, 1 'remove the starting point of upstream line
                        Pts.AddPointCollection Pts2
                        LFractalA(iA).ID = LFractalA(iA2).ID
                        LFractalA(iA).NUpstr = LFractalA(iA2).NUpstr
                        Exit For
                    End If
                End If
            Next iA2
            Exit If
        End If
    End If
Next iA

If Not FoundOneUp Then Exit Do

I = 1 + 1
If I > iMax Then
    Debug.Print "Error in DissolveLines"
    Exit Do
End If

Loop

nLinesB = 0
For iA = 1 To nLinesA
If LFractalA(iA).Temp = 0 Then
  nLinesB = nLinesB + 1
  LFractalB(nLinesB) = LFractalA(iA)
End If
Next iA
For iB = 1 To nLinesB
  LFractalA(iB) = LFractalB(iB)
Next iB
nLinesA = nLinesB

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."
End Sub

Public Sub SplitLongLines()

Debug.Print "ANGel: Entering SplitLongLines."

StartTime = Now()

nLinesB = 0

Dim xPLine As IPolyline
Dim OldPts As IPointCollection
Dim NewPts As IPointCollection
Dim pClone As IClone
Dim FoundLongLine As Boolean

ReDim LFractalB(MaxN)
Do While True
  nLinesB = 0
  FoundLongLine = False

  For iA = 1 To nLinesA
    OldFLine = LFractalA(iA)

    If OldFLine.Length > txtSplitLen.Value Then
      FoundLongLine = True
      'split
      'bottom
      nLinesB = nLinesB + 1
      Set pClone = OldFLine.PLine
      'Set pClone = LFractalA(iA).PLine
      Set xPLine = pClone.Clone
      Set OldPts = xPLine

      LFractalB(nLinesB) = xPLine

      Debug.Print "split line at nLinesB = " & nLinesB & "  length = " & xPLine.Length & "  index = " & iA
  Next iA

Debug.Print "FoundLongLine = " & FoundLongLine

Next iA

Debug.Print "Finished splitting at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."
End Sub
Set NewPts = SplitLine(OldPts, 1)
Set pClone = NewPts
Set NewFLine.PLine = pClone.Clone
NewFLine.outlet = OldFLine.outlet
NewFLine.Generation = OldFLine.Generation
NewFLine.Length = OldFLine.Length / 2
LFractalB(nLinesB) = NewFLine

'top
nLinesB = nLinesB + 1
Set NewPts = SplitLine(OldPts, 2)
Set pClone = NewPts
Set NewFLine.PLine = pClone.Clone
NewFLine.outlet = 0
NewFLine.Generation = OldFLine.Generation
NewFLine.Length = OldFLine.Length / 2
LFractalB(nLinesB) = NewFLine
Else
nLinesB = nLinesB + 1
LFractalB(nLinesB) = OldFLine
End If

Next iA

nLinesA = nLinesB

For iB = 1 To nLinesA
   LFractalA(iB) = LFractalB(iB)
Next iB

If FoundLongLine = False Then
   Exit Do
End If

Loop

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub
Public Sub Expand1Gen()

Debug.Print "ANGel: Entering Expand1Gen."

StartTime = Now()

nLinesB = 0
ReDim LFractalB(MaxN)

For iA = 1 To nLinesA

    OldFLine = LFractalA(iA)

    Dim xPLine As IPolyline
    Dim OldPts As IPointCollection
    Dim NewPts As IPointCollection
    Dim iMidIndex As Long
    Dim xb As Boolean
    Dim xi As Long
    Dim xj As Long
    Dim pClone As IClone
    'Dim xMidPoint As IPoint

    'split
    'bottom
    Set pClone = OldFLine.PLine
    Set xPLine = pClone.Clone
    ' Set xMidPoint = New Point
    Set OldPts = xPLine

    Set NewPts = SplitLine(OldPts, 1)
    Set pClone = NewPts
    Set NewFLine.PLine = pClone.Clone
    NewFLine.outlet = OldFLine.outlet
    NewFLine.Generation = OldFLine.Generation
    'LFractalB(nLinesB) = NewFLine
    nLinesB = nLinesB + 1
    'ReDim LFractalB(nLinesB)
    LFractalB(nLinesB) = NewFLine

    'top
    'Set OldPts = xPLine
    ' Set pClone = OldPts
    Set NewPts = SplitLine(OldPts, 2)
    Set pClone = NewPts
    Set NewFLine.PLine = pClone.Clone
    NewFLine.outlet = 0
    NewFLine.Generation = OldFLine.Generation
'LFractalB(nLinesB) = NewFLine
nLinesB = nLinesB + 1
'ReDim LFractalB(nLinesB)
LFractalB(nLinesB) = NewFLine

If OldFLine.NUpstr = 0 Or OldFLine.NUpstr = 1 Then

'right turn
Set NewPts = ExtendLine(OldPts, 3)
Set pClone = NewPts
Set NewFLine.PLine = pClone.Clone

Set NewFLine.PLine = New Polyline
With NewFLine.PLine
    .FromPoint = CreatePt(OldFLine.PLine.ToPoint.X, _
                            OldFLine.PLine.ToPoint.Y)
    .ToPoint = CreatePt(.FromPoint.X + dy_2 * IrrFactor(), _
                         .FromPoint.Y - dx_2 * IrrFactor())
End With
NewFLine.outlet = 0
NewFLine.Generation = k

nLinesB = nLinesB + 1
'ReDim LFractalB(nLinesB)
LFractalB(nLinesB) = NewFLine

'left turn
Set NewPts = ExtendLine(OldPts, 1)
Set pClone = NewPts
Set NewFLine.PLine = pClone.Clone

NewFLine.outlet = 0
NewFLine.Generation = k
nLinesB = nLinesB + 1
'ReDim LFractalB(nLinesB)
LFractalB(nLinesB) = NewFLine

If OldFLine.NUpstr = 0 Then
    'extension
    Set NewPts = ExtendLine(OldPts, 2)
    Set pClone = NewPts
    Set NewFLine.PLine = pClone.Clone

    NewFLine.outlet = 0
    NewFLine.Generation = k
    nLinesB = nLinesB + 1
    'ReDim LFractalB(nLinesB)
    LFractalB(nLinesB) = NewFLine
End If

End If

Next iA

nLinesA = nLinesB
ReDim LFractalA(nLinesB)

For iB = 1 To nLinesB
    LFractalA(iB) = LFractalB(iB)
Next iB

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

End Sub

Public Sub RemoveShortLines()

Debug.Print "ANGel: Entering RemoveShortlines."

StartTime = Now()

nLinesB = 0
For iA = 1 To nLinesA

    OldFLine = LFractalA(iA)

    If OldFLine.Length > xRemShortLines Then

        nLinesB = nLinesB + 1
        LFractalB(nLinesB) = OldFLine

    Else

        Debug.Print "ANGel: Shortline Removed ID = " & OldFLine.ID

    End If

Next iA

nLinesA = nLinesB
For iB = 1 To nLinesB
    LFractalA(iB) = LFractalB(iB)
Next iB
Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

End Sub
' this function takes LFractalA and writes to disk

Public Sub WriteLFractal(TempKey As String, WriteAll As Boolean)

Debug.Print "ANGel: Entering WriteLFractal."

StartTime = Now()

    ' Set up a Fields collection for the new Feature Class.
    Dim pField As IField
    Dim pFieldEdit As IFieldEdit
    Dim pFields As IFields
    Dim pFieldsEdit As IFieldsEdit
    Dim pGeomDefEdit As IGeometryDefEdit
    Dim pSR As ISpatialReference

    Set pFields = New Fields
    Set pFieldsEdit = pFields
    pFieldsEdit.FieldCount = 15

    ' Create the geometry field.
    Set pGeomDefEdit = New GeometryDef
    Set pSR = New UnknownCoordinateSystem
    With pGeomDefEdit
        .GeometryType = esriGeometryPolyline
        .HasM = False
        .HasZ = False
        Set .SpatialReference = pSR
    End With

    Set pFieldEdit = New Field
    With pFieldEdit
        .Name = "Shape"
        .AliasName = "Geometry"
        .Type = esriFieldTypeGeometry
        Set .GeometryDef = pGeomDefEdit
    End With
    Set pFieldsEdit.Field(0) = pFieldEdit

    Set pFieldEdit = New Field
    With pFieldEdit
        .Name = "ID"
        .Type = esriFieldTypeInteger
    End With
    Set pFieldsEdit.Field(1) = pFieldEdit

    Set pFieldEdit = New Field
    With pFieldEdit
.Name = "In_Node"
.Type = esriFieldTypeString
End With
Set pFieldsEdit.Field(2) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
 .Name = "Out_Node"
 .Type = esriFieldTypeString
End With
Set pFieldsEdit.Field(3) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
 .Name = "NUpstr"
 .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(4) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
 .Name = "Generation"
 .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(5) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
 .Name = "DownID"
 .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(6) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
 .Name = "Outlet"
 .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(7) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
 .Name = "Temp"
 .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(8) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
    .Name = "Orphan"
    .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(9) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
    .Name = "Length"
    .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(10) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
    .Name = "Length_ft"
    .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(11) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
    .Name = "FlowLength"
    .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(12) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
    .Name = "StrOrder"
    .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(13) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
    .Name = "StrID"
    .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(14) = pFieldEdit

'  Now create the new Shapefile. First create a Feature UID.
Dim pCLSID As UID
Set pCLSID = New UID
pCLSID.Value = "esricore.Feature"

'  Now create a new shapefile FeatureClass (check the file does not exist first).
Dim pFSO As Object
Set pFSO = CreateObject("Scripting.FileSystemObject")
If pFSO.FileExists(PathName & "L" & FileNameBase & TempKey & ".shp") Then
    MsgBox "Select different name for the new shapefile", vbInformation, "File of same name exists"
    Exit Sub
End If

Dim pFeatClass As IFeatureClass
Dim pWksp As IWorkspace
Dim pFeatWksp As IFeatureWorkspace
Dim pWkspFact As IWorkspaceFactory
Set pWkspFact = New ShapefileWorkspaceFactory
Set pFeatWksp = pWkspFact.OpenFromFile(CurDir, 0)
Set pFeatClass = pFeatWksp.CreateFeatureClass("L" & FileNameBase & TempKey & ".shp", pFields, pCLSID, Nothing, esriFTSimple, "Shape", ")

' Now, create the Line data and add it to the new FeatureClass along with the specified attributes.

If pFeatClass Is Nothing Then Exit Sub
Dim l_FCA1 As Long
Dim l_FCA2 As Long
Dim l_FCA3 As Long
Dim l_FCA4 As Long
Dim l_FCA5 As Long
Dim l_FCA6 As Long
Dim l_FCA7 As Long
Dim l_FCA8 As Long
Dim l_FCA9 As Long
Dim l_FCA10 As Long
Dim l_FCA11 As Long
Dim l_FCA12 As Long
Dim l_FCA13 As Long
Dim l_FCA14 As Long

l_FCA1 = pFeatClass.FindField("ID")
l_FCA2 = pFeatClass.FindField("ln_Node")
l_FCA3 = pFeatClass.FindField("Out_Node")
l_FCA4 = pFeatClass.FindField("NUpstr")
l_FCA5 = pFeatClass.FindField("Generation")
l_FCA6 = pFeatClass.FindField("DownID")
l_FCA7 = pFeatClass.FindField("Outlet")
l_FCA8 = pFeatClass.FindField("Temp")
l_FCA9 = pFeatClass.FindField("Orphan")
l_FCA10 = pFeatClass.FindField("Length")
l_FCA11 = pFeatClass.FindField("Length_ft")
l_FCA12 = pFeatClass.FindField("FlowLength")
l_FCA13 = pFeatClass.FindField("StrOrder")
l_FCA14 = pFeatClass.FindField("StrID")

Dim pGeomColl As IGeometryCollection
Dim pSegColl As ISegmentCollection
Dim PLine As ILine
Dim pPolyline As IPolyline
Dim pFeat As IFeature

nLinesC = 0
For iA = 1 To nLinesA
If WriteAll Or Not LFractalA(iA).Generation = 1 Then

' For each row in the Table, create a PolyLine.
Set pGeomColl = LFractalA(iA).PLine
Set pFeat = pFeatClass.CreateFeature
Set pPolyline = pGeomColl

'Set the Feature's Shape and the specified attributes.
Set pFeat.Shape = pPolyline
With LFractalA(iA)
    pFeat.Value(l_FCA1) = .ID
    pFeat.Value(l_FCA4) = .NUpstr
    pFeat.Value(l_FCA5) = .Generation
    pFeat.Value(l_FCA6) = .DownID
    pFeat.Value(l_FCA7) = .outlet
    pFeat.Value(l_FCA8) = .Temp
    pFeat.Value(l_FCA9) = .Orphan
    pFeat.Value(l_FCA10) = .Length
    pFeat.Value(l_FCA11) = .Length * 3.2808399
    pFeat.Value(l_FCA12) = .FlowLength
    pFeat.Value(l_FCA13) = .StrOrder
    pFeat.Value(l_FCA14) = .StrID
End With
pFeat.Store

Else

    nLinesC = nLinesC + 1
    LFractalC(nLinesC) = LFractalA(iA)

End If
Next iA
Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

Exit Sub

ErrorHandler:
  If Err.Number <> 0 Then
  MsgBox Err.Description, vbCritical, "Error: " & Err.Number
  End If

End Sub

Public Sub MakeBuffer()

  Debug.Print "ANGel: Entering MakeBuffer."

  StartTime = Now()

  Dim pMxDoc As IMxDocument
  Dim pMap As IMap

  Set pMxDoc = ThisDocument
  Set pMap = pMxDoc.FocusMap

  'Set up a Fields collection for the new Feature Class.
  Dim pField As IField
  Dim pFieldEdit As IFieldEdit
  Dim pFields As IFields
  Dim pFieldsEdit As IFieldsEdit

  Set pFields = New Fields
  Set pFieldsEdit = pFields
  pFieldsEdit.FieldCount = 2

  ' Create the geometry field.
  Dim pGeomDefEdit As IGeometryDefEdit
  Dim pSR As ISpatialReference
  Set pGeomDefEdit = New GeometryDef
  Set pSR = New UnknownCoordinateSystem
  With pGeomDefEdit
    .GeometryType = esriGeometryPolygon
    .HasM = False
    .HasZ = False
    Set .SpatialReference = pSR
  End With

  Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Shape"
  .AliasName = "Geometry"
  .Type = esriFieldTypeGeometry
  Set .GeometryDef = pGeomDefEdit
End With
Set pFieldsEdit.Field(0) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "ID"
  .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(1) = pFieldEdit

'Now create the new Shapefile. First create a Feature UID.
Dim pCLSID As UID
Set pCLSID = New UID
pCLSID.Value = "esricore.Feature"

'Set the input feature interfaces
Dim pInFeatureLayer As IFeatureLayer
Dim pInFeatureClass As IFeatureClass
'Use the polygon layer specified by user
Set pInFeatureLayer = getLayerByName(cboClipPoly.Value)
Set pInFeatureClass = pInFeatureLayer.FeatureClass

'Now create a new output FeatureClass for the shapefile
'(check the file does not exist first).
Dim pFSO As Object
Set pFSO = CreateObject("Scripting.FileSystemObject")
If pFSO.FileExists(PathName & "B" & FileNameBase & ".shp") Then
  MsgBox "Select different name for the new shapefile", vbInformation, 
  "File of same name exists"
  Exit Sub
End If

'Set the output feature interfaces
Dim pWkspFact As IWorkspaceFactory
Dim pWksp As IWorkspace
Dim pFeatWksp As IFeatureWorkspace
Dim pOutFeatureClass As IFeatureClass
Set pWkspFact = New ShapefileWorkspaceFactory
Set pFeatWksp = pWkspFact.OpenFromFile(CurDir, 0)
Set pOutFeatureClass = pFeatWksp.CreateFeatureClass("B" & FileNameBase & ".shp", pFields, pCLSID, Nothing, esriFTSimple, "Shape", "")
'Create input cursor and get the first feature
Dim pInFCursor As IFeatureCursor
Dim pInFeature As IFeature
Set pInFCursor = pInFeatureClass.Search(Nothing, True)
Set pInFeature = pInFCursor.NextFeature

'Perform the Buffer
' This code uses either a shapefile with one polygon, or a
'shapefile with multiple polygons, but one selected
Dim pPolygon As IPolygon
Dim pTopoOp As ITopologicalOperator
Dim pBufferGeom As IGeometry
Set pPolygon = pInFeature.Shape
Set pTopoOp = pPolygon

'Store the values in the output feature
Dim pOutFeature As IFeature
Set pOutFeature = pOutFeatureClass.CreateFeature
Set pOutFeature.Shape = pBufferGeom
pOutFeature.Value(2) = 1
pOutFeature.Store

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() -
StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Public Sub ClipLFractal()

    Debug.Print "ANGel: Entering ClipLFractal."

    StartTime = Now()
    Dim pMxDoc As IMxDocument
    Dim pWorkspaceFactory As IWorkspaceFactory
    Dim pFeatureWorkspace As IFeatureWorkspace
    Dim pInFeatureLayer As IFeatureLayer
    Dim pInFLayer As IFeatureLayer
    Dim pInFeatureClass As IFeatureClass

    Dim pClipFeatureLayer As IFeatureLayer
    Dim pClipFLayer As IFeatureLayer
    Dim pClipFeatureClass As IFeatureClass

    'Specify the workspace and the input feature class that will be clipped
Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
Set pInFeatureClass = pFeatureWorkspace.OpenFeatureClass("L" &
FILENAMEBASE & "a.shp")

' Prepare a feature layer from the above input feature class
Set pInFeatureLayer = New FeatureLayer
Set pInFeatureLayer.FeatureClass = pInFeatureClass
pInFeatureLayer.Name = pInFeatureClass.AliasName

'Specify the input feature class for clipping
Set pClipFeatureClass = pFeatureWorkspace.OpenFeatureClass("B" & FileNameBase & ".shp")

' Prepare a feature layer from the above clip feature class
Set pClipFeatureLayer = New FeatureLayer
Set pClipFeatureLayer.FeatureClass = pClipFeatureClass
pClipFeatureLayer.Name = pClipFeatureClass.AliasName

' Set output location and feature class name
Dim pNewWSName As IWorkspaceName
Set pNewWSName = New WorkspaceName
pNewWSName.WorkspaceFactoryProgID = "esriCore.ShapeFileWorkspaceFactory.1"
pNewWSName.PathName = CurDir()
Dim dirname As String
dirname = pNewWSName.PathName

Dim outfilename As String
'outfilename = pDatasetName.Name
outfilename = "L" & FileNameBase & ".shp"

'Set pDatasetName.WorkspaceName = pNewWSName

' Set the tolerance.  Passing 0.0 causes the default tolerance to be used.
' The default tolerance is 1/10,000 of the extent of the data frame's spatial domain
'Dim tol As Double
tol = txtSpaceTol.Value

Dim outName As String
outName = dirname & "\" & outfilename

' Perform the clip
'Dim pBGP As IBasicGeoprocessor
'Set pBGP = New BasicGeoprocessor
'Dim pOutputFeatClass As IFeatureClass
'Set pOutputFeatClass = pBGP.Clip(pInputTable, False, pClipTable, False, _
tol, pFeatClassName)

Dim GP As IGeoProcessor
Set GP = New GeoProcessor
Dim parameters As IVariantArray
Set parameters = New VarArray
parameters.Add pInFeatureLayer
parameters.Add pClipFeatureLayer
parameters.Add outName
GP.Execute "Clip_analysis", parameters, Nothing

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Public Sub DeleteClippedlayer()

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument

Dim pMap As IMap
Set pMap = pMxDoc.FocusMap

Dim pLayerDel As ILayer
Set pLayerDel = getLayerByName("L" & FileNameBase & "b") ' Pass the requied layer name.

If Not pLayerDel Is Nothing Then
    pMap.DeleteLayer pLayerDel
End If

End Sub

Public Sub FindOrphans()

Debug.Print "ANGel: Entering FindOrphans."

StartTime = Now()

' note: an orphan is defined as any line that traces downstream to another
' orphan or does not have a downid (except outlet)

Dim NextDownID As Integer

For iA = 1 To nLinesA
If LFractalA(iA).Orphan = 0 And LFractalA(iA).outlet = 0 Then

NextDownID = LFractalA(iA).DownID
If NextDownID = 0 Then
    LFractalA(iA).Orphan = 1
Else

    iA2 = 1
    I = 1

    Do While True

        If LFractalA(iA2).ID = NextDownID Then
            If LFractalA(iA2).Orphan = 1 _
                Or (LFractalA(iA2).DownID = 0 And Not LFractalA(iA2).outlet = 1) Then
                LFractalA(iA).Orphan = 1
                Exit Do
            End If
        End If
        If LFractalA(iA2).outlet = 1 Then
            Exit Do
        End If
        NextDownID = LFractalA(iA2).DownID
        iA2 = 1
    Else
        iA2 = iA2 + 1
    End If

    If iA2 > nLinesA Then
        Debug.Print "ANGel: Error: Can't find downstream line, assuming orphan."
        LFractalA(iA).Orphan = 1
        Exit Do
    End If

    Loop

End If

End If

Next iA

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

End Sub
Public Sub RemoveOrphans()

Debug.Print "ANGel: Entering RemoveOrphans."

StartTime = Now()

Dim NextDownID As Integer

nLinesB = 0

' Remove orphans in LFractal
For iA = 1 To nLinesA

    With LFractalA(iA)
        If .Orphan = 0 Then
            nLinesB = nLinesB + 1
            LFractalB(nLinesB) = LFractalA(iA)
        End If

    End With

Next iA

nLinesA = nLinesB
For iB = 1 To nLinesB
    LFractalA(iB) = LFractalB(iB)
Next iB

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

End Sub
Public Sub CalcFlowLength()

Debug.Print "ANGeI: Entering CalcFlowLength."

StartTime = Now()

Dim NextDownID As Integer
Dim nLinesA As Integer
End With

For iA = 1 To nLinesA

'With LFractalA(iA)

LFractalA(iA).FlowLength = LFractalA(iA).Length
NextDownID = LFractalA(iA).DownID
iA2 = 1
iA2Max = nLinesA

Do While True

If NextDownID = 0 Then Exit Do

If LFractalA(iA2).ID = NextDownID Then


If LFractalA(iA2).outlet = 1 Then Exit Do

NextDownID = LFractalA(iA2).DownID
iA2 = 1

Else

iA2 = iA2 + 1

End If

If iA2 > iA2Max Then

Debug.Print "Error: Cannot find next downstream ID in CalcFlowLength"

Exit Do

End If

Loop

'End With

Next iA

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."
End Sub

Public Sub CalcStrOrder()

Debug.Print "ANGe1: Entering CalcStrOrder."

StartTime = Now()

Dim NextDownID
Dim nUpstream As Integer
Dim iOrder As Integer
Dim nFound As Integer
Dim iOrderMax As Integer
Dim iA2Max As Integer

' --- assign all lines order 1 ---
'iOrder = 1
For iA = 1 To nLinesA
    LFractalA(iA).StrOrder = iOrder
Next iA

' --- find higher order lines, starting with 2 ---
iOrder = 2
iOrderMax = 100
Do While True

' for each line ...
'nFound = 0
For iA = 1 To nLinesA

    ' ... count the number of upstream lines with order >= current order - 1
    nUpstream = 0
    For iA2 = 1 To nLinesA
        If LFractalA(iA2).DownID = LFractalA(iA).ID Then
            If LFractalA(iA2).StrOrder >= iOrder - 1 Then
                nUpstream = nUpstream + 1
            End If
        End If
    Next iA2

    ' ... if its >= 2, incrase order of this and all downstream lines

If nUpstream $\geq 2$ Then

LFractalA(iA).StrOrder = iOrder
nFound = nFound + 1
NextDownID = LFractalA(iA).DownID
iA2 = 1
iA2Max = nLinesA

Do While True

If LFractalA(iA2).ID = NextDownID Then

LFractalA(iA2).StrOrder = iOrder
nFound = nFound + 1
NextDownID = LFractalA(iA2).DownID
iA2 = 1

Else

iA2 = iA2 + 1
End If

If NextDownID = 0 Then Exit Do
If iA2 > iA2Max Then

Debug.Print "Error: Cannot find next downstream ID in CalcStrOrder"

Exit Do
End If

Loop
End If

Next iA

If nFound = 0 Then Exit Do
iOrder = iOrder + 1
If iOrder > iOrderMax Then

Debug.Print "Error: StrOrder is too large"

Exit Do
End If

Loop

nOrder = iOrder - 1

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Public Sub CalcStrID()

Debug.Print "ANGel: Entering CalcStrID."

Dim iStrID As Long

For iA = 1 To nLinesA
LFractalA(iA).Temp = 1
For iA2 = 1 To nLinesA
    If LFractalA(iA).StrOrder = LFractalA(iA2).StrOrder And _
        LFractalA(iA).ID = LFractalA(iA2).DownID Then
        LFractalA(iA).Temp = 0
    End If
Next iA2
Next iA

iStrID = 0
Dim NextDownID As Long
Dim iA2Max As Integer

For iA = 1 To nLinesA
    'With LFractalA(iA)
    If LFractalA(iA).Temp = 1 Then
        iStrID = iStrID + 1
        LFractalA(iA).StrID = iStrID
        NextDownID = LFractalA(iA).DownID
        iA2 = 1
        iA2Max = nLinesA
        Do While True
            If NextDownID = 0 Then Exit Do
            If LFractalA(iA2).ID = NextDownID Then
                If LFractalA(iA).StrOrder = LFractalA(iA2).StrOrder Then
                    LFractalA(iA2).StrID = iStrID
                    NextDownID = LFractalA(iA2).DownID
                    iA2 = 1
                Else
                    Exit Do
                End If
            Else
                iA2 = iA2 + 1
            End If
            If iA2 > iA2Max Then
                Debug.Print "Error: Cannot find the next downstream ID in"
                CalcStrID.
                Exit Do
            End If
        Loop
    End If
End If
'End With

Next iA

nStr = iStrID

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

End Sub
Public Sub MakePFractal()

Debug.Print "ANGel: Entering MakePFractal."

Dim index As Long

StartTime = Now()

ReDim PFractalA(MaxNP)

For iA = 1 To nLinesA

    With LFractalA(iA)
        NewFPoint.X = .PLine.ToPoint.X
        NewFPoint.Y = .PLine.ToPoint.Y
        NewFPoint.ID = .ID
    End With

    PFractalA(iA) = NewFPoint

Next iA

index = iA

For iA = 1 To nLinesA

    If LFractalA(iA).DownID = 0 Then
        Debug.Print "ID of line with outlet = " & LFractalA(iA).ID
        NewFPoint.X = LFractalA(iA).PLine.FromPoint.X
        NewFPoint.ID = nLinesA + 1
        PFractalA(index) = NewFPoint
    End If

Next iA

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

' this function takes PFractalA and writes to disk

Public Sub WritePFractal(TempKey As String)

Debug.Print "ANGel: Entering WritePFractal."

StartTime = Now()
'Set up a Fields collection for the new Feature Class.
Dim pField As IField
Dim pFieldEdit As IFieldEdit
Dim pFields As IFIELDS
Dim pFieldsEdit As IFIELDSEdit
Dim pGeomDefEdit As IGeometryDefEdit
Dim pSR As ISpatialReference

Set pFields = New Fields
Set pFieldsEdit = pFields
pFieldsEdit.FieldCount = 9

' Create the geometry field.
Set pGeomDefEdit = New GeometryDef
Set pSR = New UnknownCoordinateSystem
With pGeomDefEdit
  .GeometryType = esriGeometryPoint
  .HasM = False
  .HasZ = False
  Set .SpatialReference = pSR
End With

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Shape"
  .AliasName = "Geometry"
  .Type = esriFieldTypeGeometry
  Set .GeometryDef = pGeomDefEdit
End With
Set pFieldsEdit.Field(0) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "ID"
  .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(1) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Name"
  .Type = esriFieldTypeString
End With
Set pFieldsEdit.Field(2) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "X"
End With
.Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(3) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Y"
  .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(4) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "DNode_Name"
  .Type = esriFieldTypeString
End With
Set pFieldsEdit.Field(5) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "NUp_Nodes"
  .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(6) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "DLine_Len"
  .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(7) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Inv_elev"
  .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(8) = pFieldEdit

' Now create the new Shapefile. First create a Feature UID.
Dim pCLSID As UID
Set pCLSID = New UID
pCLSID.Value = "esricore.Feature"

' Now create a new shapefile FeatureClass (check the file does not exist first).
Dim pFSO As Object
Set pFSO = CreateObject("Scripting.FileSystemObject")
If pFSO.FileExists(PathName & "P" & FileNameBase & TempKey & ".shp") Then
    MsgBox "Select different name for the new shapefile", vbInformation,
    "File of same name exists"
    Exit Sub
End If

Dim pFeatClass As IFeatureClass
Dim pWksp As IWorkspace
Dim pFeatWksp As IFeatureWorkspace
Dim pWkspFact As IWorkspaceFactory
Set pWkspFact = New ShapefileWorkspaceFactory
Set pFeatWksp = pWkspFact.OpenFromFile(CurDir, 0)
Set pFeatClass = pFeatWksp.CreateFeatureClass("P" & FileNameBase & TempKey & ".shp", pFields, pCLSID, Nothing, esriFTSimple, "Shape", "")

' Now, create the Point data and add it to the new FeatureClass along with the specified attributes.
If pFeatClass Is Nothing Then Exit Sub
Dim l_FCA1 As Long
Dim l_FCA2 As Long
Dim d1_FCA1 As Long
Dim d2_FCA1 As Long
l_FCA1 = pFeatClass.FindField("ID")
l_FCA2 = pFeatClass.FindField("Name")
d1_FCA1 = pFeatClass.FindField("X")
d2_FCA1 = pFeatClass.FindField("Y")

Dim pGeomColl As IGeometryCollection
Dim pSegColl As ISegmentCollection
Dim pPoint As IPoint
Dim pFeat As IFeature
For iA = 1 To nLinesA + 1

    ' For each row in the Table, create a PolyLine.
    Set pPoint = CreatePt(PFractalA(iA).X, PFractalA(iA).Y)
    Set pFeat = pFeatClass.CreateFeature

    ' Set the Feature’s Shape and the specified attributes.
    Set pFeat.Shape = pPoint
    With PFractalA(iA)
        pFeat.Value(l_FCA1) = .ID
        pFeat.Value(l_FCA2) = "N_" & CStr(.ID)
        pFeat.Value(d1_FCA1) = .X
        pFeat.Value(d2_FCA1) = .Y
    End With

Next iA
End With
pFeat.Store

Next iA

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

Exit Sub

ErrorHandler:
    If Err.Number <> 0 Then
        MsgBox Err.Description, vbCritical, "Error: " & Err.Number
    End If

End Sub

Sub DeleteOutlet()

    Debug.Print "ANGel: Entering DeleteOutlet."

    StartTime = Now()
    Dim pWorkspaceFactory As IWorkspaceFactory
    Dim pFeatureWorkspace As IFeatureWorkspace
    Dim pFClassPoint1 As IFeatureClass
    Dim pFClassPoint2 As IFeatureClass

        Set pWorkspaceFactory = New ShapefileWorkspaceFactory
        Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
        'Open the point fractal feature class already created
        Set pFClassPoint1 = pFeatureWorkspace.OpenFeatureClass("P" & FileNameBase & ".shp")

        Dim pDataset As IDataset
        Dim newName As String
        newName = "P" & FileNameBase & "a"
        Set pDataset = pFClassPoint1
        pDataset.Copy newName, pDataset.Workspace

        'Open the temp point fractal feature class already created
        Set pFClassPoint2 = pFeatureWorkspace.OpenFeatureClass("P" & FileNameBase & ".shp")

        Dim pQueryFilter As IQueryFilter
        Set pQueryFilter = New QueryFilter

        Dim outletID As Integer
outletID = nLinesA + 1

Dim il_FCA1 As Long
il_FCA1 = pFClassPoint2.FindField("ID")

Dim pFCursorPoint As IFeatureCursor
Set pFCursorPoint = pFClassPoint2.Update(Nothing, True)
Dim pFeaturePoint As IFeature
Set pFeaturePoint = pFCursorPoint.NextFeature

Do Until pFeaturePoint Is Nothing
    If pFeaturePoint.Value(il_FCA1) = outletID Then
        pFeaturePoint.Delete
        Exit Do
    End If
    Set pFeaturePoint = pFCursorPoint.NextFeature
Loop

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

' this function takes AFractalA and writes to disk
Public Sub WriteAFractal(TempKey As String)

    Debug.Print "ANGel: Entering WriteAFractal."

    StartTime = Now()
    ' Specify the workspace and the input point feature class
    Dim pWorkspaceFactory As IWorkspaceFactory
    Dim pFeatureWorkspace As IFeatureWorkspace
    Dim pFClassPoint As IFeatureClass
    Dim pFClassArea As IFeatureClass
    Dim pPoly1 As IPolygon
    Dim pPoly2 As IPolygon
    Dim pFCursor As IFeatureCursor
    Dim pFeatureClipPoly As IFeature
    Dim pMXDoc As IMxDocument
    Dim pActiveView As IActiveView

    Dim NewPts As IPointCollection
    Dim pClone As IClone

    Set pWorkspaceFactory = New ShapefileWorkspaceFactory
    Set pFeatureWorkspace = pWorkspaceFactory.OpenFromPath(CurDir, 0)
Set pFClassPoint = pFeatureWorkspace.OpenFeatureClass("P" & FileNameBase & ".shp")

' Set pFClassPoint = pFeatureWorkspace.OpenFeatureClass("P" & FileNameBase & ".shp")

If chkClip.Value = True Then

' Use the polygon layer specified by user in the cboClipPoly
Dim pLayerPoly As IFeatureLayer
Dim pClassPoly1 As IFeatureClass
Set pLayerPoly = getLayerByName(cboClipPoly.Value)
Set pClassPoly1 = pLayerPoly.FeatureClass
Set pCursor = pClassPoly1.Search(Nothing, False)
Set pFeatureClipPoly = pCursor.NextFeature
Set pPoly1 = pFeatureClipPoly.Shape

Else

' Use the extents of the screen display as the bounding 'Ipolygon object
Set pMxDoc = ThisDocument
Set pActiveView = pMxDoc.FocusMap
Set pClone = LFractalA(1).PLine
Set NewPts = pClone.Clone
NewPts.RemovePoints 0, NewPts.PointCount
With pActiveView.ScreenDisplay.DisplayTransformation.ConstrainedBounds
    NewPts.AddPoint CreatePt(.LowerLeft.X, .LowerLeft.Y)
    NewPts.AddPoint CreatePt(.LowerRight.X, .LowerRight.Y)
End With
Set pPoly2 = PolylineToPolygon(NewPts)

End If

Dim pGeoDataSet As IGeoDataset
Set pGeoDataSet = pFClassPoint

Dim pSR As ISpatialReference
Set pSR = pGeoDataSet.SpatialReference

' Set up a Fields collection for the new Feature Class.
Dim pField As IField
Dim pFieldEdit As IFieldeEdit
Dim pFields As IFields
Dim pFieldsEdit As IFieldsEdit
Dim pGeomDefEdit As IGeometryDefEdit
Set pFields = New Fields
Set pFieldsEdit = pFields
pFieldsEdit.FieldCount = 6

'Create the geometry field.
Set pGeomDefEdit = New GeometryDef
With pGeomDefEdit
    .GeometryType = esriGeometryPolygon
    .HasM = False
    .HasZ = False
    Set .SpatialReference = pSR
End With

Set pFieldEdit = New Field
With pFieldEdit
    .Name = "Shape"
    .AliasName = "Geometry"
    .Type = esriFieldTypeGeometry
    Set .GeometryDef = pGeomDefEdit
End With
Set pFieldsEdit.Field(0) = pFieldEdit

'Create the ID field
Set pFieldEdit = New Field
With pFieldEdit
    .Name = "ID"
    .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(1) = pFieldEdit

'Create the Area field
Set pFieldEdit = New Field
With pFieldEdit
    .Name = "Area"
    .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(2) = pFieldEdit

'Create the Area_acres field
Set pFieldEdit = New Field
With pFieldEdit
    .Name = "Area_ac"
    .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(3) = pFieldEdit

'Create the Outlet field
Set pFieldEdit = New Field
With pFieldEdit
   .Name = "Outlet"
   .Type = esriFieldTypeString
End With
Set pFieldsEdit.Field(4) = pFieldEdit

'Create the Area field
Set pFieldEdit = New Field
With pFieldEdit
   .Name = "Width_ft"
   .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(5) = pFieldEdit

Dim pFields2 As IFields2
Set pFields2 = pFClassPoint.Fields

Dim pOIDField As IField
Dim ID As Long
ID = pFields2.FindField(pFClassPoint.OIDFieldName)
Set pOIDField = pFields2.Field(ID)

'Now create the new Shapefile. First create a Feature UID.
Dim pCLSID As UID
Set pCLSID = New UID
pCLSID.Value = "esricore.Feature"

'Now create a new shapefile FeatureClass (check the file does not exist first).
Dim pFSO As Object
Set pFSO = CreateObject("Scripting.FileSystemObject")
If pFSO.FileExists(AFileName) Then
   MsgBox "Select different name for the new shapefile", vbInformation,
   "File of same name exists"
   Exit Sub
End If

Dim pOutFC As IFeatureClass
Set pOutFC = pFeatureWorkspace.CreateFeatureClass("A" & FileNameBase, pFields, pCLSID, Nothing, esriFTSimple, "Shape", "")

Dim pMap As IMap
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap
Dim pWorkspaceFact As IWorkspaceFactory
Dim pFWkSpace As IFeatureWorkspace
Set pWorkspaceFact = New ShapefileWorkspaceFactory
Set pFWkSpace = pWorkspaceFact.OpenFromFile(CurDir, 0)

If nLinesA = 1 Then

    Dim pFClass1 As IFeatureClass
    Dim pFClass2 As IFeatureClass

    Set pFClass1 = pFWkSpace.OpenFeatureClass(cboClipPoly.Value)
    Set pFClass2 = pFWkSpace.OpenFeatureClass(AFileName)

    Dim Index1 As Long
    Dim index2 As Long
    Index1 = pFClass1.FindField("AREA")
    index2 = pFClass2.FindField("Area")

    Dim NumOfRow As Long
    NumOfRow = CLng(pFClass1.FeatureCount(Nothing))

    Dim pFeature1 As IFeature
    Set pFeature1 = pFClass1.GetFeature(NumOfRow - 1)

    Dim pFeature2 As IFeature
    Set pFeature2 = pFClass2.CreateFeature
    Set pFeature2.Shape = pFeature1.Shape

    'Save the attribute of the common field
    pFeature2.Value(index2) = pFeature1.Value(Index1)
    pFeature2.Store

Else
    If chkExAShp.Value = True Then

        Dim pOldFClass As IFeatureClass
        Dim pNewFClass As IFeatureClass

        Dim pOldFeature As IFeature
        Dim pNewFeature As IFeature
        Dim pOldFCursor As IFeatureCursor
        Dim pNewFCursor As IFeatureCursor

        Set pOldFClass = pFWkSpace.OpenFeatureClass(cboExAShp.Value)
        Set pNewFClass = pFWkSpace.OpenFeatureClass(AFileName)

        'Set the two cursors to their respective feature classes
        'and to the first feature
        Set pOldFCursor = pOldFClass.Update(Nothing, True)
Set pNewFCursor = pNewFClass.Update(Nothing, True)

Set pOldFeature = pOldFCursor.NextFeature
Set pNewFeature = pNewFCursor.NextFeature
Dim A_ID_Field2 As Long
Dim j As Integer
Dim A_Area_Field2 As Long
Dim A_Areaac_Field2 As Long
'Dim A_Widthft_Field2 As Long
Dim A_Area2 As Double
A_ID_Field2 = pNewFClass.FindField("ID")
A_Area_Field2 = pNewFClass.FindField("Area")
A_Areaac_Field2 = pNewFClass.FindField("Area_ac")
'A_Widthft_Field2 = pNewFClass.FindField("Width_ft")

j = 1
'Loop through all the features in the old area feature class
Do Until pOldFeature Is Nothing

    Set pNewFeature = pNewFClass.CreateFeature
    Set pNewFeature.Shape = pOldFeature.Shape
    A_Area2 = PolygonArea(pNewFeature)
    pNewFeature.Value(A_ID_Field2) = j
    pNewFeature.Value(A_Area_Field2) = A_Area2
    pNewFeature.Value(A_Areaac_Field2) = A_Area2 * 2.471 * 10 ^ (-4)
    'pNewFeature.Value(A_Widthft_Field2) = A_Area2 ^ 0.5 * 3.2808399
    pNewFeature.Store
    Set pOldFeature = pOldFCursor.NextFeature
    Set pNewFeature = pNewFCursor.NextFeature
    j = j + 1
Loop

Else
    Set pFClassArea = pFeatureWorkspace.OpenFeatureClass(AFileName)
    Dim pTinEdit As ITinEdit
    Set pTinEdit = New Tin
    pTinEdit.InitNew pGeoDataSet.Extent

    Dim pTinAdv As ITinAdvanced2
    Set pTinAdv = pTinEdit
    pTinEdit.AddFromFeatureClass pFClassPoint, Nothing, pOIDField, pOIDField, esriTinMassPoint
    pTinEdit.Refresh

    Dim pNodeCol As ITinNodeCollection
Set pNodeCol = pTinEdit

'If nGen > 0 Then
    If chkClip.Value = True Then
        pNodeCol.ConvertToVoronoiRegions pOutFC, Nothing, pPoly1
    Else
        pNodeCol.ConvertToVoronoiRegions pOutFC, Nothing, pPoly2
    End If
End If
End If
End If

Set pFClassArea = pFeatureWorkspace.OpenFeatureClass(AFileName)

Dim P_ID_Field As Long
Dim P_Name_Field As Long
Dim A_ID_Field As Long
Dim A_Area_Field As Long
Dim A_Areaac_Field As Long
Dim A_Outlet_Field As Long
Dim A_Widthft_Field As Long

Dim pFeaturePoint As IFeature
Dim pFeatureArea As IFeature
Dim pFCursorPoint As IFeatureCursor
Dim pFCursorArea As IFeatureCursor

Dim P_ID As Long
Dim P_Name As String
Dim pPolygon As IPolygon
Dim A_Area As Double
'Dim A_Areaac As Double
Dim A_Outlet As Integer
'Dim A_Widthft As Double

' --- writing ID to AFractal and calculating and getting area ---

P_ID_Field = pFClassPoint.FindField("ID")
P_Name_Field = pFClassPoint.FindField("Name")
A_ID_Field = pFClassArea.FindField("ID")
A_Area_Field = pFClassArea.FindField("Area")
A_Areaac_Field = pFClassArea.FindField("Area_ac")
A_Outlet_Field = pFClassArea.FindField("Outlet")
'A_Widthft_Field = pFClassArea.FindField("Width_ft")

'Set the two cursors to their respective feature classes and to
'the first feature
Set pFCursorPoint = pFClassPoint.Update(Nothing, True)
Set pFCursorArea = pFClassArea.Update(Nothing, True)

Set pFeaturePoint = pFCursorPoint.NextFeature
Set pFeatureArea = pFCursorArea.NextFeature

'Loop through all the features in the point feature class
'Assign the area value to the Area field in AFractal
Do Until pFeaturePoint Is Nothing

    If pFeatureArea Is Nothing Then
        Debug.Print "ANGel/WriteAFractal: Error: Number of points
not equal to number of areas."
        Exit Do
    End If

    Set pPolygon = pFeatureArea.Value(1)
    'Assign the area value to the Area field in AFractal
    A_Area = PolygonArea(pFeatureArea)
    pFeatureArea.Value(A_Area_Field) = A_Area
    pFeatureArea.Value(A_Areaac_Field) = A_Area * 2.471 * 10 ^ (-4)
    'pFeatureArea.Value(A_Widthft_Field) = (A_Area) ^ 0.5 * 3.2808399

    'Assign the ID and name of each point to the ID and outlet
    'fields, respectively of each area in AFractal
    P_ID = pFeaturePoint.Value(P_ID_Field)
    pFeatureArea.Value(A_ID_Field) = P_ID
    'P_Name = pFeaturePoint.Value(P_Name_Field)
    'pFeatureArea.Value(A_Outlet_Field) = P_Name
    For iA = 1 To nLinesA
        If LFractalA(iA).ID = P_ID Then
            LFractalA(iA).Area = A_Area
            Exit For
        End If
    Next iA

    pFeatureArea.Store
    Set pFeaturePoint = pFCursorPoint.NextFeature
    Set pFeatureArea = pFCursorArea.NextFeature

Loop

'Release the variables
Set pGeoDataSet = Nothing
Set pSR = Nothing
Set pFields = Nothing
Set pOIDField = Nothing
Set pOutFC = Nothing
Set pNodeCol = Nothing

ErrorHandler:
    If Err.Number <> 0 Then
    MsgBox Err.Description, vbCritical, "Error: " & Err.Number
    End If

    Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Public Function PolygonArea(pFeature As IFeature) As Double
    Dim pArea As IArea
    If pFeature.Shape.GeometryType = esriGeometryPolygon Then
        Set pArea = pFeature.Shape
        PolygonArea = pArea.Area
        Exit Function
    End If
End Function

' this function takes LFractalA and writes to disk
Public Sub WriteLFractal(TempKey As String, WriteAll As Boolean)

    Debug.Print "ANGel: Entering WriteLFractal."

    StartTime = Now()

    ' Set up a Fields collection for the new Feature Class.
    Dim pField As IField
    Dim pFieldEdit As IFieldEdit
    Dim pFields As IFields
    Dim pFieldsEdit As IFieldsEdit
    Dim pGeomDefEdit As IGeometryDefEdit
    Dim pSR As ISpatialReference

    Set pFields = New Fields
    Set pFieldsEdit = pFields
    pFieldsEdit.FieldCount = 15

    ' Create the geometry field.
    Set pGeomDefEdit = New GeometryDef
Set pSR = New UnknownCoordinateSystem
With pGeomDefEdit
  .GeometryType = esriGeometryPolyline
  .HasM = False
  .HasZ = False
  Set .SpatialReference = pSR
End With

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Shape"
  .AliasName = "Geometry"
  .Type = esriFieldTypeGeometry
  Set .GeometryDef = pGeomDefEdit
End With
Set pFieldsEdit.Field(0) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "ID"
  .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(1) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "In_Node"
  .Type = esriFieldTypeString
End With
Set pFieldsEdit.Field(2) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Out_Node"
  .Type = esriFieldTypeString
End With
Set pFieldsEdit.Field(3) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "NUpstr"
  .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(4) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Generation"
End With
.Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(5) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "DownID"
  .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(6) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Outlet"
  .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(7) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Temp"
  .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(8) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Orphan"
  .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(9) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Length"
  .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(10) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
  .Name = "Length_ft"
  .Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(11) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
.Name = "FlowLength"
.Type = esriFieldTypeDouble
End With
Set pFieldsEdit.Field(12) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
 .Name = "StrOrder"
 .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(13) = pFieldEdit

Set pFieldEdit = New Field
With pFieldEdit
 .Name = "StrID"
 .Type = esriFieldTypeInteger
End With
Set pFieldsEdit.Field(14) = pFieldEdit

' Now create the new Shapefile. First create a Feature UID.
Dim pCLSID As UID
Set pCLSID = New UID
pCLSID.Value = "esricore.Feature"

' Now create a new shapefile FeatureClass (check the file does not exist first).
Dim pFSO As Object
Set pFSO = CreateObject("Scripting.FileSystemObject")
If pFSO.FileExists(PathName & "L" & FileNameBase & TempKey & ".shp") Then
 MsgBox "Select different name for the new shapefile", vbInformation, "File of same name exists"
 Exit Sub
End If

Dim pFeatClass As IFeatureClass
Dim pWksp As IWorkspace
Dim pFeatWksp As IFeatureWorkspace
Dim pWkspFact As IWorkspaceFactory
Set pWkspFact = New ShapefileWorkspaceFactory
Set pFeatWksp = pWkspFact.OpenFromFile(CurDir, 0)
Set pFeatClass = pFeatWksp.CreateFeatureClass("L" & FileNameBase & TempKey & ".shp", pFields, pCLSID, Nothing, esriFTSimple, "Shape", "")

' Now, create the Line data and add it to the new FeatureClass along with the specified attributes.
If pFeatClass Is Nothing Then Exit Sub
Dim l_FCA1 As Long
Dim l_FCA2 As Long
Dim l_FCA3 As Long
Dim l_FCA4 As Long
Dim l_FCA5 As Long
Dim l_FCA6 As Long
Dim l_FCA7 As Long
Dim l_FCA8 As Long
Dim l_FCA9 As Long
Dim l_FCA10 As Long
Dim l_FCA11 As Long
Dim l_FCA12 As Long
Dim l_FCA13 As Long
Dim l_FCA14 As Long

l_FCA1 = pFeatClass.FindField("ID")
l_FCA2 = pFeatClass.FindField("In_Node")
l_FCA3 = pFeatClass.FindField("Out_Node")
l_FCA4 = pFeatClass.FindField("NUpstr")
l_FCA5 = pFeatClass.FindField("Generation")
l_FCA6 = pFeatClass.FindField("DownID")
l_FCA7 = pFeatClass.FindField("Outlet")
l_FCA8 = pFeatClass.FindField("Temp")
l_FCA9 = pFeatClass.FindField("Orphan")
l_FCA10 = pFeatClass.FindField("Length")
l_FCA11 = pFeatClass.FindField("Length_ft")
l_FCA12 = pFeatClass.FindField("FlowLength")
l_FCA13 = pFeatClass.FindField("StrOrder")
l_FCA14 = pFeatClass.FindField("StrID")

Dim pGeomColl As IGeometryCollection
Dim pSegColl As ISegmentCollection
Dim PLine As ILine
Dim pPolyline As IPolyline
Dim pFeat As IFeature

nLinesC = 0
For iA = 1 To nLinesA
If WriteAll Or Not LFractalA(iA).Generation = 1 Then

' For each row in the Table, create a PolyLine.
Set pGeomColl = LFractalA(iA).PLine
Set pFeat = pFeatClass.CreateFeature
Set pPolyline = pGeomColl

'Set the Feature's Shape and the specified attributes.

Set pFeat.Shape = pPolyline
With LFractalA(iA)
    pFeat.Value(l_FCA1) = .ID
    pFeat.Value(l_FCA4) = .NUpstr
    pFeat.Value(l_FCA5) = .Generation
    pFeat.Value(l_FCA6) = .DownID
    pFeat.Value(l_FCA7) = .outlet
    pFeat.Value(l_FCA8) = .Temp
    pFeat.Value(l_FCA9) = .Orphan
    pFeat.Value(l_FCA10) = .Length
    pFeat.Value(l_FCA11) = .Length * 3.2808399
    pFeat.Value(l_FCA12) = .FlowLength
    pFeat.Value(l_FCA13) = .StrOrder
    pFeat.Value(l_FCA14) = .StrID
End With
pFeat.Store

Else

    nLinesC = nLinesC + 1
    LFractalC(nLinesC) = LFractalA(iA)

End If
Next iA

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

Exit Sub

ErrorHandler:
    If Err.Number <> 0 Then
        MsgBox Err.Description, vbCritical, "Error: " & Err.Number
    End If

End Sub
Public Sub CalcDrainageLineAttributes()

Debug.Print "ANGel: Entering CalcDrainageLineAttributes."

StartTime = Now()
Dim pWorkspaceFactory As IWorkspaceFactory
Dim pFeatureWorkspace As IFeatureWorkspace
Dim pFClassLine As IFeatureClass

Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
'Open the polyline and the polygon feature classes already created
'Set pFClassLine = pFeatureWorkspace.OpenFeatureClass("DL" & FileNameBase & ".shp")
Set pFClassLine = pFeatureWorkspace.OpenFeatureClass("L" & FileNameBase & ".shp")

Dim i1_FCA1 As Long
Dim i2_FCA1 As Long
Dim i3_FCA1 As Long
Dim i4_FCA1 As Long
Dim i5_FCA1 As Long
'Dim d1_FCA1 As Long

i1_FCA1 = pFClassLine.FindField("ID")
i2_FCA1 = pFClassLine.FindField("In_Node")
i3_FCA1 = pFClassLine.FindField("Out_Node")
i4_FCA1 = pFClassLine.FindField("DownID")
i5_FCA1 = pFClassLine.FindField("NUpstr")

Dim pFCursorline1 As IFeatureCursor
Dim pFCursorline2 As IFeatureCursor
Dim pFeatureline1 As IFeature
Dim pFeatureline2 As IFeature

Set pFCursorline1 = pFClassLine.Update(Nothing, True)
Set pFeatureline1 = pFCursorline1.NextFeature
Set pFCursorline2 = pFClassLine.Update(Nothing, True)
Set pFeatureline2 = pFCursorline2.NextFeature

Dim pPline1 As IPolyline
Dim pPline2 As IPolyline
Dim FoundDownID As Boolean

'Find the downstream ID of each drainage line
Do Until pFeatureline1 Is Nothing
    Set pPline1 = pFeatureline1.Shape
    FoundDownID = False
'Debug.Print "F1 " & pFeatureline1.Value(2)

Do Until pFeatureline2 Is Nothing
    Set pPline2 = pFeatureline2.Shape
    If ((pPline1.FromPoint.X - pPline2.ToPoint.X) ^ 2 + (pPline1.FromPoint.Y - pPline2.ToPoint.Y) ^ 2) ^ 0.5 < SpaceTol Then
        pFeatureline1.Value(i4_FCA1) = pFeatureline2.Value(i1_FCA1)
        FoundDownID = True
        pFeatureline1.Store
    End If
    ' take F2 to next
    Set pFeatureline2 = pFCursorline2.NextFeature
Loop ' end of F2
If FoundDownID = False Then
    pFeatureline1.Value(i4_FCA1) = 0
    pFeatureline1.Store
End If

'Take F2 back to the first feature

Set pFCursorline2 = pFClassLine.Update(Nothing, True)
Set pFeatureline2 = pFCursorline2.NextFeature
Set pPline2 = pFeatureline2.Shape
Set pFeatureline1 = pFCursorline1.NextFeature
Loop ' end of F1

'Calculate the number of upstream drainage lines
Set pFCursorline1 = pFClassLine.Update(Nothing, True)
Set pFeatureline1 = pFCursorline1.NextFeature
Set pFCursorline2 = pFClassLine.Update(Nothing, True)
Set pFeatureline2 = pFCursorline2.NextFeature

Dim NUpstr As Integer

Do Until pFeatureline1 Is Nothing
    Set pPline1 = pFeatureline1.Shape
    NUpstr = 0
    'Debug.Print "F1 " & pFeatureline1.Value(2)
    Do Until pFeatureline2 Is Nothing
        Set pPline2 = pFeatureline2.Shape
        If ((pPline1.ToPoint.X - pPline2.FromPoint.X) ^ 2 + (pPline1.ToPoint.Y - pPline2.FromPoint.Y) ^ 2) ^ 0.5 < SpaceTol Then
            NUpstr = NUpstr + 1
        End If
    End If
End If
' take F2 to next
Debug.Print "NUpstr before F2 set to next feature = " & NUpstr
Set pFeatureline2 = pFCursorline2.NextFeature
Loop ' end of F2
pFeaturelinel.Value(i5_FCA1) = NUpstr
pFeaturelinel.Store
'Take F2 back to the first feature
Set pFCursorline2 = pFCursorline2.Update(Nothing, True)
Set pFeatureline2 = pFCursorline2.NextFeature
Set pline2 = pFeatureline2.Shape
Set pFeaturelinel = pFeaturelinel.NextFeature
Loop ' end of F1
Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."
End Sub
Sub CalcNodeID()
    Debug.Print "ANGel: Entering CalcNodeID."
    StartTime = Now()
    Dim pWorkspaceFactory As IWorkspaceFactory
    Dim pFeatureWorkspace As IFeatureWorkspace
    Dim pFClassLine As IFeatureClass
    Dim pFClassPoint As IFeatureClass
    Dim pFCursorLine As IFeatureCursor
    Dim pFCursorPoint As IFeatureCursor
    Dim pFeatureLine As IFeature

    Set pWorkspaceFactory = New ShapefileWorkspaceFactory
    Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
    'Open the polyline feature class already created
    'Set pFClassLine = pFeatureWorkspace.OpenFeatureClass("DL" & FileNameBase & ".shp")
    Set pFClassLine = pFeatureWorkspace.OpenFeatureClass("L" & FileNameBase & ".shp")
    'Open the point feature class already created
    'Set pFClassPoint = pFeatureWorkspace.OpenFeatureClass("DP" & FileNameBase & ".shp")
    Set pFClassPoint = pFeatureWorkspace.OpenFeatureClass("P" & FileNameBase & ".shp")

    Dim pFCursorLine As IFeatureCursor
    Dim pFCursorPoint As IFeatureCursor
    Dim pFeatureLine As IFeature
Dim pFeaturePoint As IFeature

Set pFCursorLine = pFClassLine.Update(Nothing, True)
Set pFeatureLine = pFCursorLine.NextFeature
Set pFCursorPoint = pFClassPoint.Update(Nothing, True)
Set pFeaturePoint = pFCursorPoint.NextFeature

Dim sLIn_NodeName As Long
Dim sLOut_NodeName As Long
Dim sP_NodeName As Long

sLIn_NodeName = pFClassLine.FindField("In_Node")
sLOut_NodeName = pFClassLine.FindField("Out_Node")
sP_NodeName = pFClassPoint.FindField("Name")

Dim pPline As IPolyline
Dim pPoint As IPoint

' Dim FoundDownID As Boolean

' Find the downstream ID of each drainage line
Do Until pFeatureLine Is Nothing
    Set pPline = pFeatureLine.Shape
    Do Until pFeaturePoint Is Nothing
        Set pPoint = pFeaturePoint.Shape
        If ((pPline.ToPoint.X - pPoint.X) ^ 2 + _(pPline.ToPoint.Y - pPoint.Y) ^ 2) ^ 0.5 < SpaceTol Then
            pFeatureLine.Value(sLIn_NodeName) = pFeaturePoint.Value(sP_NodeName)
            pFeatureLine.Store
        End If
        ' If pPline.FromPoint.X = pPoint.X And pPline.FromPoint.Y = pPoint.Y Then
        If ((pPline.FromPoint.X - pPoint.X) ^ 2 + _(pPline.FromPoint.Y - pPoint.Y) ^ 2) ^ 0.5 < SpaceTol Then
            pFeatureLine.Value(sLOut_NodeName) = pFeaturePoint.Value(sP_NodeName)
            pFeatureLine.Store
        End If

        ' take F2 to next
        Set pFeaturePoint = pFCursorPoint.NextFeature
    Loop ' end of F2

Set pFCursorPoint = pFClassPoint.Update(Nothing, True)
Set pFeaturePoint = pFCursorPoint.NextFeature
Set pPoint = pFeaturePoint.Shape
Sub CalcNodeInvElev()
    Debug.Print "ANGel: Entering CalcNodeInvElev."
    StartTime = Now()
    Dim pWorkspaceFactory As IWorkspaceFactory
    Dim pFeatureWorkspace As IFeatureWorkspace
    Dim pFClassLine As IFeatureClass
    Dim pFClassPoint As IFeatureClass

    Set pWorkspaceFactory = New ShapefileWorkspaceFactory
    Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
    Set pFClassLine = pFeatureWorkspace.OpenFeatureClass("L" & FileNameBase & ".shp")
    Set pFClassPoint = pFeatureWorkspace.OpenFeatureClass("P" & FileNameBase & ".shp")

    Dim pFCursorLine As IFeatureCursor
    Dim pFCursorPoint As IFeatureCursor
    Dim pFeatureLine As IFeature
    Dim pFeaturePoint As IFeature

    Set pFCursorLine = pFClassLine.Update(Nothing, True)
    Set pFeatureLine = pFCursorLine.NextFeature
    Set pFCursorPoint = pFClassPoint.Update(Nothing, True)
    Set pFeaturePoint = pFCursorPoint.NextFeature

    Dim pPoint As IPoint
    Dim pPline As IPolyline

    Dim sNode_Name As Long
    Dim sDownNode_Name As Long
    Dim iNUp_Nodes As Long
    Dim dLLength_ft As Long
    Dim dLine_Len_ft As Long
    Dim dNodeInv_Elev As Long
    Dim sLIn_NodeName As Long
    Dim sLOut_NodeName As Long
Dim FoundDownNode As Boolean

sNode_Name = pFClassPoint.FindField("Name")
sDownNode_Name = pFClassPoint.FindField("DNode_Name")
iNUp_Nodes = pFClassPoint.FindField("NUp_Nodes")

' This line to be used when only LFractal is generated
dLLength_ft = pClassLine.FindField("Length_ft")

' This line to be used when DLFractal is generated
'dLLength_ft = pClassLine.FindField("Length")
dLine_Len_ft = pFClassPoint.FindField("DLine_Len")
dNodeInv_Elev = pFClassPoint.FindField("Inv_elev")
sLIn_NodeName = pFClassLine.FindField("In_Node")
sLOut_NodeName = pFClassLine.FindField("Out_Node")

'Determine the downstream node name for each point
Do Until pFeaturePoint Is Nothing
    Set pPoint = pFeaturePoint.Shape
    FoundDownNode = False

    Do Until pFeatureLine Is Nothing
        Set pPline = pFeatureLine.Shape

        If ((pPline.ToPoint.X - pPoint.X) ^ 2 + (pPline.ToPoint.Y - pPoint.Y) ^ 2) ^ 0.5 < SpaceTol Then
            pFeaturePoint.Value(dLine_Len_ft) = pFeatureLine.Value(dLLength_ft)
            pFeaturePoint.Value(sDownNode_Name) = pFeatureLine.Value(sLOut_NodeName)
            FoundDownNode = True
        End If

        ' take F2 to next
        Set pFeatureLine = pFeatureLine.NextFeature

    End If

    ' take F2 to next
    Set pFeatureLine = pFeatureLine.NextFeature

Loop ' end of F2

If FoundDownNode = False Then
    pFeaturePoint.Value(sDownNode_Name) = "N_0"
pFeaturePoint.Value(dLine_Len_ft) = 0
    pFeaturePoint.Value(dNodeInv_Elev) = 0
    pFeaturePoint.Store
End If

'Take F2 back to the first feature

Set pFCursorLine = pClassLine.Update(Nothing, True)
Set pFeatureLine = pFCursorLine.NextFeature

' Set pPline = pFeatureLine.Shape
Set pFeaturePoint = pFCursorPoint.NextFeature
Loop ' end of F1

'Calculate the number of upstream drainage lines
Set pFCursorPoint = pFClassPoint.Update(Nothing, True)
Set pFeaturePoint = pFCursorPoint.NextFeature
Set pFCursorLine = pFClassLine.Update(Nothing, True)
Set pFeatureLine = pFCursorLine.NextFeature

Dim NUpstr_Nodes As Integer

Do Until pFeaturePoint Is Nothing
    Set pPoint = pFeaturePoint.Shape
    NUpstr_Nodes = 0
    Do Until pFeatureLine Is Nothing
        Set pPline = pFeatureLine.Shape
        If ((pPoint.X - pPline.FromPoint.X)^2 + (pPoint.Y - pPline.FromPoint.Y)^2)^0.5 < SpaceTol Then
            NUpstr_Nodes = NUpstr_Nodes + 1
        End If
    Set pFeatureLine = pFCursorLine.NextFeature
Loop ' end of F2
pFeaturePoint.Value(iNUp_Nodes) = NUpstr_Nodes
pFeaturePoint.Store

'Take F2 back to the first feature

Set pFCursorLine = pFClassLine.Update(Nothing, True)
Set pFeatureLine = pFCursorLine.NextFeature
'Set pPline = pFeatureLine.Shape
Set pFeaturePoint = pFCursorPoint.NextFeature
Loop ' end of F1

Dim pFCursorPoint1 As IFeatureCursor
Set pFCursorPoint1 = pFClassPoint.Update(Nothing, True)
Dim pFeaturePoint1 As IFeature
Set pFeaturePoint1 = pFCursorPoint1.NextFeature

Dim pFCursorPoint2 As IFeatureCursor
Dim pFeaturePoint2 As IFeature
Set pFCursorPoint2 = pFClassPoint.Update(Nothing, True)
Set pFeaturePoint2 = pFCursorPoint2.NextFeature

Dim dNode_Name As String
Dim count As Integer
Dim countmax As Integer
Dim FoundOneZero As Boolean

count = 1
countmax = 100 'xxx needs fix

'This do loop calculates the node invert elevation for all
'the nodes until there is only one node (the outlet) with zero
'invert elevation

Do While True

    Debug.Print "This is the " & count & "th iteration in the do while true loop"
    FoundOneZero = False

    Do Until pFeaturePoint1 Is Nothing
        If pFeaturePoint1.Value(dNodeInv_Elev) = 0 And pFeaturePoint1.Value(sDownNode_Name) <> "N_0" Then
            dNode_Name = pFeaturePoint1.Value(sDownNode_Name)
            Do Until pFeaturePoint2 Is Nothing
                If pFeaturePoint2.Value(sNode_Name) = dNode_Name Then
                    'Debug.Print "F2 Matched with " & pFeaturePoint2.Value(sNode_Name)
                    If pFeaturePoint2.Value(dNodeInv_Elev) <> 0 Or pFeaturePoint2.Value(sDownNode_Name) = "N_0" Then
                        'Calculate the invert elevation based on the input value_
                        'for pipe slope if enetered by user, or else based on_
                        'an average pipe slope value
                        FoundOneZero = True

                        If chkAvgPipeSlp.Value = True Then
                        Else
                            pFeaturePoint1.Value(dNodeInv_Elev) = pFeaturePoint2.Value(dNodeInv_Elev) + (0.013 * pFeaturePoint1.Value(dLine_Len_ft))
                        End If
                        pFeaturePoint1.Store
                        'Debug.Print "F1 " & pFeaturePoint1.Value(sNode_Name) & " has inv_elev = " & pFeaturePoint1.Value(dNodeInv_Elev)
                    End If
                End If
                pFeaturePoint2.Store
            End Do
        End If
    End Do

End If
End If
Set pFeaturePoint2 = pFCursorPoint2.NextFeature
Loop
Set pFCursorPoint2 = pFClassPoint.Update(Nothing, True)
Set pFeaturePoint2 = pFCursorPoint2.NextFeature
' Debug.Print " after update getting reset to " & pFeaturePoint2.Value(sNode_Name)
End If

Set pFeaturePoint1 = pFCursorPoint1.NextFeature
Loop
Set pFCursorPoint1 = pFClassPoint.Update(Nothing, True)
Set pFeaturePoint1 = pFCursorPoint1.NextFeature

Dim pQueryFilter As IQueryFilter
Set pQueryFilter = New QueryFilter

' set the where clause
pQueryFilter.SubFields = "Inv_elev"
pQueryFilter.WhereClause = "Inv_elev = 0"

' If Not FoundOneZero Then Exit Do

' Get a count of all the features
If pFClassPoint.FeatureCount(pQueryFilter) = 1 Then Exit Do
' End If

count = count + 1

If count > 400 Then Exit Do
' End If

Loop

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

End Sub

Sub FindOutletforEachSubcatchment()

Debug.Print "ANGel: Entering FindOutletforEachSubcatchment."

StartTime = Now()
' Get the focused map from MapDocument
Dim pWorkspaceFactory As IWorkspaceFactory

Dim pFeatureWorkspace As IFeatureWorkspace
Dim pFClassPoint As IFeatureClass
Dim pFClassArea As IFeatureClass

Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
Set pFClassPoint = pFeatureWorkspace.OpenFeatureClass("P" & FileNameBase & ".shp")
Set pFClassArea = pFeatureWorkspace.OpenFeatureClass("A" & FileNameBase & ".shp")

Dim pFeaturePoint As IFeature
Dim pFeatureArea As IFeature
Dim pFCursorPoint As IFeatureCursor
Dim pFCursorArea As IFeatureCursor

'Set the two cursors to their respective feature classes and to the first feature
Set pFCursorPoint = pFClassPoint.Update(Nothing, True)
Set pFCursorArea = pFClassArea.Update(Nothing, True)

Set pFeaturePoint = pFCursorPoint.NextFeature
Set pFeatureArea = pFCursorArea.NextFeature

Dim P_Name_Field As Long
Dim P_DNodeName_Field As Long
Dim A_Outlet_Field As Long

P_Name_Field = pFClassPoint.FindField("Name")
P_DNodeName_Field = pFClassPoint.FindField("DNode_Name")
A_Outlet_Field = pFClassArea.FindField("Outlet")

Dim P_Name As String
Dim A_Outlet As String

'Loop through all the features in the area feature class and the point feature class and assign the node that is contained within an area completely as its outlet
Set pFCursorArea = pFClassArea.Update(Nothing, True)
Set pFeatureArea = pFCursorArea.NextFeature

Dim pSQ As ISpatialFilter
Dim outlet As String
Dim pFCurPoint1 As IFeatureCursor
Dim pFCurPoint2 As IFeatureCursor
Dim pFPoint1 As IFeature
Dim pFPoint2 As IFeature

Do Until pFeatureArea Is Nothing
    Set pSQ = New SpatialFilter
    pSQ.GeometryField = pFClassArea.ShapeFieldName
    Set pSQ.Geometry = pFeatureArea.Shape

    'If there are no points within the extents of the polygon, get the points that intersect the polygon

    'set the spatial relationship
    pSQ.SpatialRel = esriSpatialRelContains
    If pFClassPoint.FeatureCount(pSQ) <> 0 Then
        Set pFCursorPoint = pFClassPoint.Search(pSQ, True)
        Set pFeaturePoint = pFCursorPoint.NextFeature

        outlet = pFeaturePoint.Value(P_Name_Field)
        pFeatureArea.Value(A_Outlet_Field) = outlet
    Else
        pSQ.SpatialRel = esriSpatialRelIntersects
        Set pFCurPoint1 = pFClassPoint.Search(pSQ, True)
        Set pFPoint1 = pFCurPoint1.NextFeature
        Set pFCurPoint2 = pFClassPoint.Search(pSQ, True)
        Set pFPoint2 = pFCurPoint2.NextFeature

        Do Until pFPoint1 Is Nothing
            Do Until pFPoint2 Is Nothing
                If pFPoint1.Value(P_Name_Field) = pFPoint2.Value(P_DNodeName_Field) Then
                    'get the "Name" field from the point shapefile
                    outlet = pFPoint1.Value(P_Name_Field)
                    pFeatureArea.Value(A_Outlet_Field) = outlet
                End If
                Set pFPoint2 = pFCurPoint2.NextFeature
            Loop
            Set pFCurPoint2 = pFClassPoint.Search(pSQ, True)
        Set pFPoint2 = pFCurPoint2.NextFeature
        'Debug.Print " after update getting reset to " & pFeaturePoint2.Value(sNode_Name)
    End If
    Set pFPoint1 = pFCurPoint1.NextFeature
    Loop
    Set pFCurPoint1 = pFClassPoint.Search(pSQ, True)
    Set pFPoint1 = pFCurPoint2.NextFeature
End If
pFeatureArea.Store
Set pFeatureArea = pFCursorArea.NextFeature
Loop

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Public Sub ZonalStatistics(readGrid As String, SWMMParam As String)
'Get the focused map from MapDocument
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pMap As IMap
Set pMap = pMxDoc.FocusMap

'Get the zone dataset (feature or raster) from the directory
Dim pAFLayer As IFeatureLayer
Set pAFLayer = getLayerByName("A" & FileNameBase)
Dim pPFLayer As IFeatureLayer
Set pPFLayer = getLayerByName("P" & FileNameBase)

Dim pZoneGeoDataset As IGeoDataset
Dim sFieldName As String
sFieldName = "FID"

'Create a FeatureClassDescriptor from featureclass zone dataset
'If TypeOf pLayer Is IFeatureLayer Then
' Dim pFLayer As IFeatureLayer
' Set pFLayer = pLayer

Dim pAFClass As IFeatureClass
Set pAFClass = pAFLayer.FeatureClass
Dim pPFClass As IFeatureClass
Set pPFClass = pPFLayer.FeatureClass

Dim pFDescriptor As IFeatureClassDescriptor
Set pFDescriptor = New FeatureClassDescriptor
pFDescriptor.Create pAFClass, Nothing, sFieldName
Set pZoneGeoDataset = pFDescriptor

'Get the value raster from the form
Dim pValueRasLayer As IRasterLayer
Set pValueRasLayer = getRasLayerByName(readGrid)
Dim pValueRaster As IRaster
Set pValueRaster = pValueRasLayer.Raster

'Create a RasterZonalOp operator
Dim pZonalOp As IZonalOp
Set pZonalOp = New RasterZonalOp

'Set output workspace
Dim pEnv As IRasterAnalysisEnvironment
Set pEnv = pZonalOp
Dim pWS As IWorkspace
Dim pWSF As IWorkspaceFactory
Set pWSF = New RasterWorkspaceFactory
Set pWS = pWSF.OpenFromFile(CurDir, 0)
Set pEnv.OutWorkspace = pWS

'Create zonal statistics table
Dim pTable As ITable
Dim pOutputRaster As IGeoDataset
'Set pTable = pZonalOp.ZonalStatisticsTable(pZoneGeoDataset, pValueRaster, esriGeoAnalysisStatsMean, True)
Set pOutputRaster = pZonalOp.ZonalStatistics(pZoneGeoDataset, pValueRaster, esriGeoAnalysisStatsMean, True)

Dim pOutputLayer As IRasterLayer
Set pOutputLayer = New RasterLayer
pOutputLayer.CreateFromRaster pOutputRaster
'pOutputLayer.Name = "Slope_ZonalMean"
'pMap.AddLayer pOutputLayer

Call AddRasterToAreaFractal(pOutputRaster, pAFClass, SWMMParam)
End Sub

Sub AddRasterToAreaFractal(pInRaster As IRaster, pInFeatureClass As IFeatureClass, sFieldName As String)
    ' pInRaster: input raster
    ' pInFeatureClass: input point feature class
    ' sFieldName: name of the field that stores the values

    'On Error GoTo ERH

    Debug.Print "ANGel: Entering AddRastertoAreaFractal."

    ' Define field name
    Dim pFld As IFIELDEDIT
    Set pFld = New FIELD

pFld.Name = sFieldName

' Define field type
Dim pProp As IRasterProps
Set pProp = pInRaster
If pProp.PixelType = PT_CHAR Or pProp.PixelType = PT_UCHAR Then
    pFld.Type = esriFieldString
    pFld.Length = 20
    pFld.Required = 0
ElseIf pProp.PixelType = PT_FLOAT Or pProp.PixelType = PT_DOUBLE Or pProp.PixelType Then
    pFld.Type = esriFieldDouble
    pFld.Length = 24
    pFld.Required = 8
Else ' for integer case
    pFld.Type = esriFieldInteger
    pFld.Length = 24
    pFld.Required = 0
End If

'Add Field
pInFeatureClass.AddField pFld

'Debug.Print "line after error"

' Get field index
Dim FieldIndex As Integer
FieldIndex = pInFeatureClass.FindField(sFieldName)
If FieldIndex < 0 Then Exit Sub

' Create a raster layer and QI for IIdentify interface
Dim pRLayer As IRasterLayer
Set pRLayer = New RasterLayer
pRLayer.CreateFromRaster pInRaster
Dim pIdentify As IIdentify
Set pIdentify = pRLayer

Dim pIDArray As IArray
Dim pRIDObj As IRasterIdentifyObj
Dim I As Long
Dim pArea As IArea
'Dim pPoint As IPoint
Dim pFeature As IFeature
'Dim pNewArea As IArea
Dim pNewPoint As IPoint
'Set pNewArea = New Area
Set pNewPoint = New Point
'Loop through each area in the feature class and obtain value of the
'raster on the label point within that area

Dim NumOfRow As Integer
'Dim pArea_Ac As Double
NumOfRow = pInFeatureClass.FeatureCount(Nothing)
For I = 0 To NumOfRow - 1
  'Get point
  Set pFeature = pInFeatureClass.GetFeature(I)
  Set pArea = pFeature.Shape
  'Set pPoint = pFeature.Shape
  pNewPoint.X = pArea.LabelPoint.X
  pNewPoint.Y = pArea.LabelPoint.Y
  'pNewPoint.X = pArea.Centroid.X

  'Get RasterIdentifyObject on that point
  Set pIDArray = pIdentify.Identify(pNewPoint)
  If Not pIDArray Is Nothing Then
    Set pRIDObj = pIDArray.Element(0)
    'Get the value of the RasterIdentifyObject and add it to the
    field
    If pProp.PixelType = PT_CHAR Or pProp.PixelType = PT_UCHAR Then
      pFeature.Value(FieldIndex) = pRIDObj.Name
    ElseIf pProp.PixelType = PT_FLOAT Or pProp.PixelType = PT_DOUBLE Or pProp.PixelType Then
      If pRIDObj.Name <> "NoData" Then
        pFeature.Value(FieldIndex) = CDbl(pRIDObj.Name)
      End If
      Else  ' for integer case
        If pRIDObj.Name <> "NoData" Then
          pFeature.Value(FieldIndex) = CLng(pRIDObj.Name)
        End If
      End If
    pFeature.Store
  End If
Next I
'Exit Sub
'ERH:
'   MsgBox Err.Description
  Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() -
StartTime) * 24# * 60# * 60, 1)) & " seconds."
End Sub

Public Sub CalcWidth()
Debug.Print "ANGel: Entering CalcWidth"

StartTime = Now()
Dim pWorkspaceFactory As IWorkspaceFactory
Dim pFeatureWorkspace As IFeatureWorkspace
Dim pFClassArea As IFeatureClass

Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
'Set pFClassLine = pFeatureWorkspace.OpenFeatureClass(\"DL\" & FileNameBase & ".shp")
Set pFClassArea = pFeatureWorkspace.OpenFeatureClass(\"A" & FileNameBase & ".shp\")

Dim i1_FCA1 As Long
Dim i2_FCA1 As Long
Dim i3_FCA1 As Long
Dim i4_FCA1 As Long

i1_FCA1 = pFClassArea.FindField("Area_ac")
i2_FCA1 = pFClassArea.FindField("LOF")
i3_FCA1 = pFClassArea.FindField("Width_ft")
i4_FCA1 = pFClassArea.FindField("ID")

Dim pFCursorArea As IFeatureCursor
Dim pFeatureArea As IFeature

Set pFCursorArea = pFClassArea.Update(Nothing, True)
Set pFeatureArea = pFCursorArea.NextFeature

Dim pPline1 As IPolyline
Dim pPline2 As IPolyline
Dim FoundDownID As Boolean

'Calculate the width of each area from its area divided by its_ 'overland flow length
Do Until pFeatureArea Is Nothing
  Debug.Print "division by zero in line " & pFeatureArea.Value(i4_FCA1)
  pFeatureArea.Value(i3_FCA1) = pFeatureArea.Value(i1_FCA1) * 43560 / pFeatureArea.Value(i2_FCA1)
  pFeatureArea.Store
  'take cursor to next
  Set pFeatureArea = pFCursorArea.NextFeature
Loop ' end of loop
'Take cursor back to the first feature

    Set pFCursorArea = pFClassArea.Update(Nothing, True)
    Set pFeatureArea = pFCursorArea.NextFeature

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."

End Sub
Public Sub AddPShapeFile()

    Debug.Print "ANGel: Entering AddPShapeFile."

    StartTime = Now()
    Dim pMxDoc As IMxDocument
    Dim pMap As IMap
    Dim pWorkspaceFactory As IWorkspaceFactory
    Dim pFeatureWorkspace As IFeatureWorkspace
    Dim pFeatureLayer As IFeatureLayer
    Dim pFeatureClass As IFeatureClass

    'Specify the workspace and the feature class
    Set pWorkspaceFactory = New ShapefileWorkspaceFactory
    Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
    Set pFeatureClass = pFeatureWorkspace.OpenFeatureClass("P" & FileNameBase & ".shp")

    'Prepare a feature layer
    Set pFeatureLayer = New FeatureLayer
    Set pFeatureLayer.FeatureClass = pFeatureClass
    pFeatureLayer.Name = pFeatureLayer.FeatureClass.AliasName

    'Add the feature layer to the active map
    Set pMxDoc = ThisDocument
    Set pMap = pMxDoc.FocusMap
    pMap.AddLayer pFeatureLayer

    'Refresh the active view
    pMxDoc.ActivatedView.Refresh

    Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() -
    StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Public Sub AddAShapeFile()

    Debug.Print "ANGel: Entering AddAShapeFile."

    StartTime = Now()
    Dim pMxDoc As IMxDocument
    Dim pMap As IMap
    Dim pWorkspaceFactory As IWorkspaceFactory
    Dim pFeatureWorkspace As IFeatureWorkspace
    Dim pFeatureLayer As IFeatureLayer
    Dim pFeatureClass As IFeatureClass


'Specify the workspace and the feature class
Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
Set pFeatureClass = pFeatureWorkspace.OpenFeatureClass("A" & FileNameBase & ".shp")

'Prepare a feature layer
Set pFeatureLayer = New FeatureLayer
Set pFeatureLayer.FeatureClass = pFeatureClass
pFeatureLayer.Name = pFeatureLayer.FeatureClass.AliasName

'Add the feature layer to the active map
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap
pMap.AddLayer pFeatureLayer

'Refresh the active view
pMxDoc.ActivatedView.Refresh

Debug.Print "Finished at " & Now() & ": after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Public Sub AddLShapeFile()

Debug.Print "ANGel: Entering AddLShapeFile."

StartTime = Now()
Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pWorkspaceFactory As IWorkspaceFactory
Dim pFeatureWorkspace As IFeatureWorkspace
Dim pFeatureLayer As IFeatureLayer
Dim pFeatureClass As IFeatureClass

'Specify the workspace and the feature class
Set pWorkspaceFactory = New ShapefileWorkspaceFactory
Set pFeatureWorkspace = pWorkspaceFactory.OpenFromFile(CurDir, 0)
Set pFeatureClass = pFeatureWorkspace.OpenFeatureClass(LFileName)

'Prepare a feature layer
Set pFeatureLayer = New FeatureLayer
Set pFeatureLayer.FeatureClass = pFeatureClass
pFeatureLayer.Name = pFeatureLayer.FeatureClass.AliasName

'Add the feature layer to the active map
Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap
pMap.AddLayer pFeatureLayer

'Refresh the active view
pMxDoc.ActivatedView.Refresh

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Public Sub ColorLShapeFile()

Debug.Print "ANGel: Entering ColorLShapeFile."

StartTime = Now()
'Part 1: Define a fractal color ramp
Dim FractalColorRamp (10, 2) As Long

FractalColorRamp(1, 0) = 255 'Red
FractalColorRamp(2, 1) = 59
FractalColorRamp(3, 2) = 20

FractalColorRamp(2, 0) = 14 'Green
FractalColorRamp(2, 1) = 214
FractalColorRamp(2, 2) = 0

FractalColorRamp(3, 0) = 20 'Blue
FractalColorRamp(3, 1) = 48
FractalColorRamp(3, 2) = 255

FractalColorRamp(4, 0) = 145 'Purple
FractalColorRamp(4, 1) = 0
FractalColorRamp(4, 2) = 194

FractalColorRamp(5, 0) = 255 'Yellow
FractalColorRamp(5, 1) = 255
FractalColorRamp(5, 2) = 0

FractalColorRamp(6, 0) = 0 'Cyan
FractalColorRamp(6, 1) = 207
FractalColorRamp(6, 2) = 214

FractalColorRamp(7, 0) = 161 'Dark Umber
FractalColorRamp(7, 1) = 51
FractalColorRamp(7, 2) = 37
Part 2: Prepare a unique value renderer

Dim pUniqueValueRenderer As IUniqueValueRenderer
Dim maxsymcount As Integer
Dim pSym As ILineSymbol
Dim ingen As Long

maxsymcount = nOrder

' Define the renderer
Set pUniqueValueRenderer = New UniqueValueRenderer
pUniqueValueRenderer.FieldCount = 1
pUniqueValueRenderer.Field(0) = "StrOrder"

' Get the input layer and feature class.
Dim pFLayer As IFeatureLayer
Set pFLayer = getLayerByName("L" & FileNameBase)

Dim pFClass As IFeatureClass
Set pFClass = pFLayer.FeatureClass
Dim pFields As IFields
Set pFields = pFClass.Fields

For ingen = 1 To maxsymcount
    If ingen > 11 Then
        Set pSym = New SimpleLineSymbol
        pSym.Color = GetRGBColor(255, 85, 0)  ' for generations higher than 10, all lines will be of same color (fir red)
        pSym.Width = 11
        pUniqueValueRenderer.AddValue Trim(Str(ingen)), "StrOrder", pSym
    Else
        Set pSym = New SimpleLineSymbol
        pSym.Color = GetRGBColor(FractalColorRamp(ingen, 0), FractalColorRamp(ingen, 1), FractalColorRamp(ingen, 2))
pSym.Width = ingen
pUniqueValueRenderer.AddValue Trim(Str(ingen)), "StrOrder", pSym

End If

Next ingen

'Part3: Assign the renderer to the feature layer
'and refresh the map and its table of contents

Dim pMxDoc As IMxDocument
Dim pMap As IMap
Dim pFeatureLayer As IFeatureLayer
Dim pGeoFeatureLayer As IGeoFeatureLayer

Set pMxDoc = ThisDocument
Set pMap = pMxDoc.FocusMap

'Assign the renderer to the feature layer
Set pGeoFeatureLayer = pFLayer
Set pGeoFeatureLayer.Renderer = pUniqueValueRenderer
'Refresh the map and its table of contents
pMxDoc.ActiveView.PartialRefresh esriViewGeography, pFeatureLayer, Nothing
pMxDoc.UpdateContents

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24# * 60# * 60, 1)) & " seconds."

End Sub

Function GetRGBColor(R As Long, G As Long, B As Long)
  Dim pColor As IRgbColor
  Set pColor = New RgbColor

  pColor.Red = R
  pColor.Green = G
  pColor.Blue = B
  GetRGBColor = pColor

End Function

'-----------------------------
'--- calculate statistics ---
Public Sub CalcStats()

Debug.Print "ANGel: Entering CalcStats."
StartTime = Now()

' --- open file & write header ---
Open PathName & FileNameBase & ".txt" For Output As #1
Print #1, "ANGel: Artificial Network Generator"
Print #1, "Network Statistics"
Print #1, "Note: All in map units."
Print #1, "Generated on " & Now()
Print #1, ""

' --- basic line & area statistics ---
Dim Line_Length_Sum As Double
Dim Line_Length_Ave As Double
Dim Line_FlowLength_Sum As Double
Dim Line_FlowLength_Ave As Double
Dim Line_Area_Sum As Double
Dim Line_Area_Ave As Double

Line_Length_Sum = 0#
Line_FlowLength_Sum = 0#
Line_Area_Sum = 0#

iA = 0

For iA = 1 To nLinesA
    With LFractalA(iA)
        Line_Length_Sum = Line_Length_Sum + .Length
        Line_FlowLength_Sum = Line_FlowLength_Sum + .FlowLength
    End With
End If
If txtUserGen.Value = 0 Then
    Dim pFLayerPoly As IFeatureLayer
    Dim pFClassPoly1 As IFeatureClass
    Dim pFeatureClipPoly As IFeature
    Dim pFCursor As IFeatureCursor
    Dim pArea As IArea

Set pFLayerPoly = getLayerByName(cboClipPoly.Value)
Set pFClassPoly1 = pFLayerPoly.FeatureClass
Set pFCursor = pFClassPoly1.Search(Nothing, False)
Set pFeatureClipPoly = pFCursor.NextFeature
Set pArea = pFeatureClipPoly.Shape
Line_Area_Sum = pArea.Area

Else

    Line_Area_Sum = Line_Area_Sum + LFractalA(iA).Area

End If

Next iA

Line_Length_Ave = Line_Length_Sum / nLinesA
Line_FlowLength_Ave = Line_FlowLength_Sum / nLinesA
Line_Area_Ave = Line_Area_Sum / nLinesA

Print #1, "Basic Line & Area Statistics"
Print #1, "Number of Lines = " & Trim(Str(nLinesA))
Print #1, "Sum, Ave Line Length = " & Trim(Str(Line_Length_Sum)) & ", " & Trim(Str(Line_Length_Ave))
Print #1, "Sum, Ave Line FlowLength = " & Trim(Str(Line_FlowLength_Sum)) & ", " & Trim(Str(Line_FlowLength_Ave))
Print #1, "Sum, Ave Line Area = " & Trim(Str(Line_Area_Sum)) & ", " & Trim(Str(Line_Area_Ave))
Print #1, ""

',
' --- basic stream statistics ---
',

Dim Str_Length() As Double
ReDim Str_Length(nStr)
Dim Str_Order() As Double
ReDim Str_Order(nStr)

Dim Str_Length_Sum As Double
Dim Str_Length_Ave As Double

For iA = 1 To nLinesA
    With LFractalA(iA)
        Str_Length(.StrID) = Str_Length(.StrID) + .Length
        Str_Order(.StrID) = .StrOrder
    End With
Next iA

Str_Length_Sum = 0#
Dim iStr As Long

For iStr = 1 To nStr
    Str_Length_Sum = Str_Length_Sum + Str_Length(iStr)
Next iStr

Str_Length_Ave = Str_Length_Sum / nStr

Print #1, "Basic Stream Statistics"
Print #1, "Number of Streams = " & Trim(Str(nStr))
Print #1, "Sum, Ave Stream Length = " & Trim(Str(Str_Length_Sum)) & ", " & Trim(Str(Str_Length_Ave))
Print #1, ""

' --- hydrology (e.g. horton) statistics ---

Dim Horton_N() As Long
Dim Horton_r_Sum() As Double
Dim Horton_r_Ave() As Double
Dim Horton_Rb_Sum As Double
Dim Horton_Rr_Ave As Double
Dim Horton_Rr_Sum As Double
Dim DrainageDensity As Double

ReDim Horton_N(nOrder)
ReDim Horton_r_Sum(nOrder)
ReDim Horton_r_Ave(nOrder)

'Horton_N(Str_Order(iStr)) represents the number of streams of order (Str_Order(iStr))
'Horton_r_Sum(Str_Order(iStr)) represents the sum of the lengths of streams of order (Str_Order(iStr))
iStr = 0
For iStr = 1 To nStr
    Horton_N(Str_Order(iStr)) = Horton_N(Str_Order(iStr)) + 1
    Horton_r_Sum(Str_Order(iStr)) = Horton_r_Sum(Str_Order(iStr)) + Str_Length(iStr)
Next iStr
Next iStr

Dim iG As Integer
Horton_Rb_Sum = 0#

If nOrder = 1 Then
    DrainageDensity = Line_Length_Sum / Line_Area_Sum
Print #1, "Hydrology (e.g. Horton) Statistics"
Print #1, "Bifurcation Ratio (Rb) cannot be calculated for single order streams"
Print #1, "Length-Order Ratio (Rr) cannot be calculated for single order streams"
Print #1, "Drainage Density = " & Trim(Str(DrainageDensity))
Print #1, ""
Else

For iG = 1 To nOrder - 1
    Horton_Rb_Sum = Horton_Rb_Sum + Horton_N(iG) / Horton_N(iG + 1)
    Horton_Rr_Sum = Horton_Rr_Sum + (Horton_r_Sum(iG + 1) / Horton_N(iG + 1)) / (Horton_r_Sum(iG) / Horton_N(iG))
 Next iG
Horton_Rb_Ave = Horton_Rb_Sum / (nOrder - 1)
Horton_Rr_Ave = Horton_Rr_Sum / (nOrder - 1)

DrainageDensity = Line_Length_Sum / Line_Area_Sum

Print #1, "Hydrology (e.g. Horton) Statistics"
Print #1, "Bifurcation Ratio (Rb) = " & Trim(Str(Horton_Rb_Ave))
Print #1, "Length-Order Ratio (Rr) = " & Trim(Str(Horton_Rr_Ave))
Print #1, "Drainage Density = " & Trim(Str(DrainageDensity))
Print #1, ""
End If

' --- fractal statistics ---

Dim Fractal_D As Double

If nOrder = 1 Then
    Print #1, "Fractal Statistics"
    Print #1, "Fractal Dimension cannot be calculated for single order streams."
    Print #1, ""
Else
    Fractal_D = Log(Horton_Rb_Ave) / Log(Horton_Rr_Ave)

    Print #1, "Fractal Statistics"
    Print #1, "Fractal Dimension (D) = " & Trim(Str(Fractal_D))
    Print #1, ""
End If

' --- close file ---
Close #1

Debug.Print "Finished at " & Now() & " after " & Trim(Round((Now() - StartTime) * 24 * 60 * 60, 1)) & " seconds."
End Sub