

**EPISODIC PROCESSES IN FLUVIAL LANDSCAPE  
EVOLUTION**

by

**Simon Dadson**

B.A. (Hons), University of Oxford, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF

**Master of Science**

in

THE FACULTY OF GRADUATE STUDIES

(Department of Geography)

We accept this thesis as conforming  
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

September 2000

© Simon Dadson, 2000

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of GEOGRAPHY

The University of British Columbia  
Vancouver, Canada

Date 6 SEPTEMBER 2000

## Abstract

The fractal dimension provides a succinct summary of topographic intricacy and a measure of landscape roughness. The work presented in this thesis is an attempt to examine the spatial variation of fractal properties of topography. It is hypothesized that distinct zones, or domains, may be identified in the landscape, within which the fractal dimension is constant, owing to the dominance of a characteristic geomorphic process.

The variogram method (structure-function analysis) is used to measure the fractal properties of digital elevation models of the Queen Charlotte Islands and small tributaries of the Capilano River (in the Southern Coast Mountains). Process domains are identified using upslope-area criteria, and are shown to have distinct fractal dimensions. In the Capilano basin, the fractal dimension is found to be higher on upper slopes; whereas on the Queen Charlotte Islands, it is higher on lower slopes. This difference is explained with reference to the details of specific surficial processes operating in each of the areas, and to their different glacial histories.

A multiple-process landscape evolution model is presented to explore the link between geomorphic process and fractal form. The model is carefully calibrated using field data from several recent surveys within the study areas. Non-linear diffusion is used to represent surficial slope failure; fluvial processes are represented using a simplified Bagnold equation; and deep-seated rock slides are introduced at random on the steepest, highest slopes. The model results confirm the smoothing effect of diffusion, which lowers the fractal dimension of the resulting topography. Structures introduced by rock-slope failures introduce visible short-term complexity on hillslopes; however, this is quickly incorporated into the channel network so does not appear to increase the fractal dimension of areas defined as hillslopes using the threshold slope-area criterion. Further work is suggested to improve modelling of fluvial processes in steep, headwater channels, and to investigate the details of hillslope-channel coupling at higher spatial resolution.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>Acknowledgements</b>	<b>xi</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Aims . . . . .	1
1.1.1 Spatial Organization of Landscape Processes . . . . .	2
1.2 Fractal Models of Fluvial Landscapes . . . . .	5
1.2.1 Fractal Properties of Topography . . . . .	5
1.3 Objectives . . . . .	11
<b>Chapter 2 Surface Processes and Tectonic History in the Queen Charlotte Islands and Southern Coast Mountains of British Columbia</b>	<b>12</b>
2.1 Location . . . . .	12
2.2 Tectonic background and Geological History . . . . .	16
2.2.1 Tectonics and Topography . . . . .	16
2.3 Queen Charlotte Islands . . . . .	17
2.3.1 Bedrock Geology . . . . .	18

2.3.2	Glacial Legacy . . . . .	18
2.3.3	Contemporary Processes . . . . .	19
2.4	GVRD Capilano Watershed . . . . .	20
2.4.1	Bedrock Geology . . . . .	20
2.4.2	Tectonic Uplift . . . . .	21
2.4.3	Glacial Legacy . . . . .	21
2.4.4	Contemporary Processes . . . . .	22
2.5	Summary . . . . .	23
 <b>Chapter 3 Sediment Flux due to Episodic Events</b>		<b>24</b>
3.1	Process rates . . . . .	24
3.2	Sources of Data . . . . .	24
3.3	Surficial Slides: magnitude-frequency . . . . .	25
3.3.1	Results . . . . .	25
3.3.2	Power-law Scaling and Self-organized Criticality . . . . .	26
3.3.3	Power-law Scaling and Sampling Bias . . . . .	28
3.3.4	Finite Scaling Effects . . . . .	28
3.3.5	GVRD Slides . . . . .	31
3.3.6	Implications . . . . .	32
3.4	Slope and Volumetric Transport Rate . . . . .	33
3.4.1	Approach . . . . .	33
3.4.2	Parameterization . . . . .	33
3.4.3	Discussion . . . . .	36
3.5	Rock Slides . . . . .	36
3.5.1	Rock Slide Inventory . . . . .	37
3.5.2	Discussion . . . . .	37

3.6	Summary . . . . .	39
<b>Chapter 4 Fractal properties of topography</b>		<b>40</b>
4.1	Geomorphic Processes and Fractal Landscapes . . . . .	40
4.2	Fractal Properties of Landscape . . . . .	41
4.3	Measuring Fractal Properties of Landscape . . . . .	43
4.4	Process-domains and the Fractal Dimension . . . . .	44
4.5	Results . . . . .	46
4.5.1	GVRD Capilano Watershed . . . . .	46
4.5.2	Queen Charlotte Islands . . . . .	60
4.6	Discussion . . . . .	69
<b>Chapter 5 Synthesis: Process-Based Modelling with Episodic Events</b>		<b>72</b>
5.1	Introduction . . . . .	72
5.2	Constructing Testable Landscape Models . . . . .	73
5.3	Framework for Landscape Modelling . . . . .	74
5.3.1	Continuum Model of Landscape Evolution . . . . .	75
5.3.2	Parameters . . . . .	78
5.3.3	Canonical Scaling . . . . .	80
5.3.4	Boundary Conditions . . . . .	81
5.3.5	Numerical Schemes . . . . .	81
5.3.6	Stability . . . . .	82
5.4	Results: Preliminary Runs . . . . .	83
5.5	Results: Detailed Model Runs . . . . .	88
5.5.1	Lembke Creek . . . . .	89
5.5.2	Gregory Creek . . . . .	94
5.6	Discussion . . . . .	95

<b>Chapter 6 Conclusion</b>	<b>98</b>
6.1 Practical Implications . . . . .	99
6.2 Theoretical Implications . . . . .	100
<b>Bibliography</b>	<b>102</b>
<b>Appendix A Computation of Fractal Dimension</b>	<b>112</b>
A.1 Slope-Area Plot Constructions . . . . .	112
A.2 Construction of Hillslope Variograms . . . . .	116
<b>Appendix B Derivation of Landscape Modelling Equations</b>	<b>132</b>
B.1 Continuity Equation . . . . .	132
B.2 Fluvial Transport Rule . . . . .	133
B.3 erode2d.cc . . . . .	135

## List of Tables

2.1	Sediment Flux due to Landsliding on Queen Charlotte Islands (data from Rood 1984) . . . . .	20
3.1	Slope-Volume Classes for GVRD Landslides . . . . .	34
3.2	Parameters for Non-Linear Slope-Volume Relation . . . . .	34
4.1	GVRD Capilano Channel Network Fractal Dimensions . . . . .	47
4.2	GVRD Capilano Hillslope Fractal Statistics . . . . .	59
4.3	Drainage Basin Characteristics and Channel Network Fractal Dimensions: Queen Charlotte Islands . . . . .	63
4.4	Queen Charlotte Islands Hillslope Fractal Statistics . . . . .	68
5.1	Parameter Estimates for Landscape Evolution Model . . . . .	79
5.2	Parameters Used in Model Runs . . . . .	88
5.3	Fractal Dimensions of Model Results . . . . .	90

## List of Figures

1.1	Continuum of Geomorphic Processes (after Montgomery and Foufoula-Georgiou, 1993). The thick solid line is typical slope-area plot for a steepland landscape; the thick broken line is a typical plot for a lowland area. . . . .	3
1.2	Simulated Fractal Landscapes: fractional Brownian surfaces (Klinkenberg and Goodchild, 1993) . . . . .	6
1.3	Moment statistics of a land-surface (part of Daniels Creek in the Capilano Watershed; see Chapter Four). The lower of the two plots shows a linear moment-order relation, indicating that this particular landscape is a mono-fractal . . . . .	9
2.1	Location of the two Study Areas (also showing principal tectonic features, described in this chapter) . . . . .	13
2.2	Queen Charlotte Islands (study basins are described here and in Chapter Four) . . . . .	14
2.3	GVRD Capilano Watershed (study basins are described here and in Chapter Four) . . . . .	15
3.1	Queen Charlotte Islands: Landslide Magnitude Frequency . . . . .	26
3.2	Extreme Event Distribution . . . . .	29

3.3	Resolution Function . . . . .	30
3.4	Landslide Magnitude Frequency modified by Resolution Function . . . . .	30
3.5	GVRD Slope-Volume Relation . . . . .	35
3.6	Rock Slide Magnitude Frequency (data from Savigny 1991) . . . . .	38
4.1	Zones for Computation of $D$ . . . . .	48
4.2	Slope-Area Plots: GVRD Daniels Creek . . . . .	49
4.3	Slope-Area Plots: GVRD Sisters Creek . . . . .	50
4.4	Slope-Area Plots: GVRD Enchantment Creek . . . . .	51
4.5	Hillslope Variograms: GVRD Daniels Creek (D1 – D2) . . . . .	53
4.6	Hillslope Variograms: GVRD Daniels Creek (D3 – D4) . . . . .	54
4.7	Hillslope Variograms: GVRD Sisters Creek (S1 – S2) . . . . .	55
4.8	Hillslope Variograms: GVRD Sisters Creek (S3 – S4) . . . . .	56
4.9	Hillslope Variograms: GVRD Enchantment Creek (E1 – E2) . . . . .	57
4.10	Hillslope Variograms: GVRD Enchantment Creek (E2 – E3) . . . . .	58
4.11	Slope-Area Plots: Queen Charlotte Islands (Rennel Sound Area) . . . . .	61
4.12	Slope-Area Plots: Queen Charlotte Islands (Skidegate Channel Area) . . . . .	62
4.13	Hillslope Variograms: Queen Charlotte Islands (Rennel Sound: Riley and Gregory Creeks) . . . . .	64
4.14	Hillslope Variograms: Queen Charlotte Islands (Rennel Sound: Moun- tain and Hangover Creeks) . . . . .	65
4.15	Hillslope Variograms: Queen Charlotte Islands (Skidegate Channel: Ar- mentieres and Government Creeks) . . . . .	66
4.16	Hillslope Variograms: Queen Charlotte Islands (Skidegate Channel: Ja- son and Security Creeks) . . . . .	67
5.1	Artificial Model Surface after 100ka . . . . .	86

5.2	Model Rock Slides: Magnitude-Frequency (100 year slide interval; N = 1000) . . . . .	87
5.3	Model Rock Slides: Magnitude-Frequency (400 year slide interval; N = 250) . . . . .	87
5.4	Lembke Creek: 10 ka no rock slides . . . . .	91
5.5	Lembke Creek: 10 ka with rock slides . . . . .	92
5.6	Gregory Creek: 10 ka with rock slides . . . . .	96

## Acknowledgements

I am grateful for the assistance and encouragement given to me by Michael Church, who supervised this work throughout. Olav Slaymaker made several suggestions during the formulation of ideas for this work and gave friendly encouragement. Brian Klinkenberg provided help with GIS, and invaluable discussions of fractal concepts. Garry Clarke provided excellent help with the formulation and implementation of the landscape evolution model described in Chapter Five; this numerical model has benefitted from insights obtained by reference to the recent work of Yvonne Martin (acknowledged in the text), which have greatly speeded its implementation.

I am grateful to Russell White for sharing some of the benefit of his field experience in the Capilano watershed. Darren Ham helped with ARC/INFO and ArcView GIS software. Catherine Griffiths gave several computer graphics tutorials. Vincent Kujala generously provided access to computing facilities and gave assistance with the  $\text{\LaTeX}$  typesetting software.

The analyses presented here would have been impossible had not the following people and organizations provided access to data: Tony Cheong (Queen Charlotte Islands DEM data); Derek Bonin, Dave Dunkley, Lorne Gilmour (Greater Vancouver Regional District DEM and landslide data); Ken Rood (Queen Charlotte Islands landslide data); Wayne Savigny (Fraser Valley landslide data).

This work was completed while the author was recipient of the Wendy Fan Memorial Scholarship and a University Graduate Fellowship at the University of British Columbia. Finally, I am grateful to Elaine Cho and the office staff, who made the University easier to deal with, and to my family and Emma McKenzie for their support.

SIMON DADSON

*The University of British Columbia*

*August 2000*

# Chapter 1

## Introduction

### 1.1 Aims

Studies of fluvial landscape evolution have demonstrated that fluvial networks create fractal landscapes<sup>1</sup> (see Rodríguez-Iturbe and Rinaldo, 1997, for a recent, comprehensive review). The intuitive justification for this result is that branching river networks dissect landscapes over a wide range of scales. The effects of hillslope processes such as rock slides, shallow landslides and debris flows on the fractal dimension of the landscape have been less well studied. The aim of the present work is to determine the extent to which hillslope processes modify the effects of dissection by fractal fluvial networks.

In order to achieve this goal, the rates and spatio-temporal variability of episodic, structure-generating hillslope processes are examined in conjunction with an analysis of the spatial variation of the fractal dimension in different process-domains within the landscape. Finally, an empirically calibrated model of fluvial and hillslope processes is used to make explicit the links between episodic processes and the fractal dimension of

---

<sup>1</sup>A preliminary, non-technical definition of a fractal is a shape that is composed of parts of itself, with each part in some way similar to the whole. A fractal landscape is therefore a landscape that shows similar variability at all spatial scales. This definition is elaborated in the following sections.

topography. With these objectives in mind, there follows some theoretical justification for the link between geomorphic processes and fractal landscapes.

### 1.1.1 Spatial Organization of Landscape Processes

The properties of topography are determined by geomorphic processes; it is therefore reasonable to expect that spatial variations in the fractal dimension of topography will be related to spatial variations in landscape-forming processes. Several dynamical considerations may be used to explain the spatial organization of landscape processes. Fluvial landscapes comprise water and differentially erodible sediment in spatially and temporally variable quantities. The relative amounts of water and sediment lead to different geomorphic processes, which cause changes in the configuration of the landscape. Geomorphic processes have been organized according to relative amounts of water and sediment by Takahashi (1981), Savage (1993), and Montgomery and Foufoula-Georgiou (1993) *inter alia*. The scheme of Montgomery and Foufoula-Georgiou is the most recent and most pertinent to the present study area, so it will form a prototype for the discussion presented here.

Montgomery and Foufoula-Georgiou's scheme (illustrated in Figure 1.1) puts various fluvial and hillslope processes into categories controlled by upslope area (serving as a proxy for both the amount and persistence of hydrological inputs), and slope. They suggest that it is instructive to consider hillslope and fluvial transport as end-members of a continuum of landscape processes ranging from turbulent fluvial washload transport, through bedload transport and gravity-driven processes in low-order streams, to shallow surficial hillslope failures and soil creep. Upon this range of processes it is necessary to impose other processes, such as deep-seated rockslides, which are contingent on the local configuration of bedrock features, and pervasive processes such as chemical denudation (by solution).

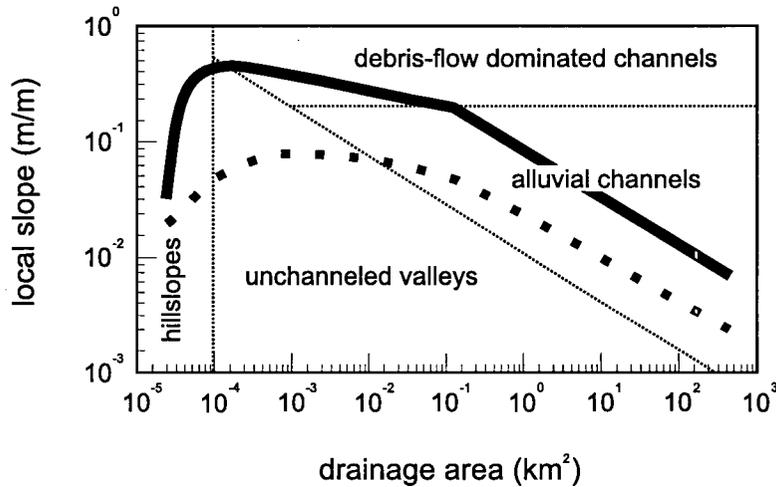


Figure 1.1: Continuum of Geomorphic Processes (after Montgomery and Foufoula-Georgiou, 1993). The thick solid line is typical slope-area plot for a steepland landscape; the thick broken line is a typical plot for a lowland area.

These modifications notwithstanding, one may expect to find testable evidence in the landscape for this interpretation of the fundamental controls on sediment transporting processes, and therefore landscape form. Several authors have outlined empirically-justified schemes for ordering landscape process and their associated morphological changes in this way (Ryder, 1981; Culling, 1988; Hewitt, 1989).

The relation between altitudinally-zoned processes has often stood aside from altitudinally-zoned form (as observed by Outcalt and Melton, 1992; Evans and McClean, 1995); their re-amalgamation is central to the attempt within this thesis to relate fractal properties of topography to the processes that produce them. Among writers who *do* discuss the relation between the spatio-temporal structure of process and form at the landscape scale, few discuss the *fractal* properties of form. However Veneziano and Iacobellis (1999) consider fractal properties in some detail: they quickly point out that the scaling properties of hillslope and fluvial processes are not the same. They suggest that channel networks are fractal because channel incision varies as a

fractional power of upslope area, which takes values over a range of scales between zero and the size of the area of interest, leading to changes in the landscape with no characteristic scale. Furthermore, Veneziano and Iacobellis (1999) claim that fractal behaviour should not be expected on hillslopes because it is not inherent in the governing equation for hillslope evolution, which is usually supposed to be gradient-driven linear diffusion.

Culling (1960), Kirkby (1971), Tucker and Slingerland (1994), Martin and Church (1997) and Martin (1998) have shown that diffusion is a good representation of hillslope *processes* over long time-scales, however it may not be adequate at reproducing the properties of landscape *form* at all spatial scales. In particular, diffusion does not correctly reproduce the fractal properties of real landscapes (as discussed explicitly by Turcotte, 1998).

Diffusion can be used to represent processes that can be averaged over timescales of interest (Martin and Church, 1997); it therefore correctly reproduces the average changes of a landscape. However, observations of form (e.g., digital elevation data) are instantaneous samples that reveal the effects of events that occur below the timescales over which diffusion is valid. If these events (e.g., landslides) are not smooth, they might be expected to create landscape zones with fractal dimensions greater than two (i.e., non-planar hillslopes). Indeed, Hovius *et al.* (1997), Pelletier *et al.* (1997), Densmore *et al.* (1997) and Turcotte (1999) have shown that individual mass-movement processes responsible for hillslope evolution are power-law distributed (over a limited range of scales), which suggests that fractal scaling ought to be an inherent feature of hillslope morphology. In order to present these arguments in greater detail, it is necessary to define the concept of a fractal landscape in a more rigorous way.

## 1.2 Fractal Models of Fluvial Landscapes

### 1.2.1 Fractal Properties of Topography

The fractal dimension of a topographic surface is a measure of the degree to which elevation changes over different spatial scales. It is an indicator of landscape intricacy and an indirect measure of landscape roughness (Outcalt and Melton, 1992).

A fractal is a shape without a characteristic length scale. It has a non-integer Hausdorff (fractal<sup>2</sup>) dimension,  $D$ , indicating that it is a space-filling curve (if  $1 < D < 2$ ), or a volume-filling surface (if  $2 < D < 3$ ). As geometric objects, fractals are characterized by the fact that they are invariant under scale transformations (enlargement or reduction). Examples of fractional Brownian surfaces with fractal dimensions between two and three are shown in Figure 1.2.

The history of the use of fractal concepts to characterize geomorphic features begins with L. F. Richardson<sup>3</sup>, who showed that coastlines vary over many length scales. Later, writing about branching river networks, Horton (1945) presented laws showing relations between statistics of streams of various sizes. None of these authors' works were interpreted in the context of fractal geometry until Mandelbrot (1967) pointed out the utility of this erstwhile theoretical branch of pure mathematics for describing intricate features of the natural world, which he elaborated considerably in later work<sup>4</sup>. His *Fractal Geometry of Nature* (1983) presents several geomorphological examples of fractal geometry including branching river networks, intricate island coasts and topographic transects.

---

<sup>2</sup>The Hausdorff and fractal dimensions are not always equivalent. Exceptions to the equivalence assumed here are uncommon, so will not be discussed (see Barnsley, 1988, for more rigorous definitions of each of these measures).

<sup>3</sup>described in Mandelbrot (1983).

<sup>4</sup>While Mandelbrot was the first to use the term fractal (*fractus* (Lat.) having been broken), various mathematicians (e.g., von Koch, Peano, Sierpinski) had studied the theoretical properties of curves that would now be called fractals (see Mandelbrot, 1983, for historical perspectives on this).

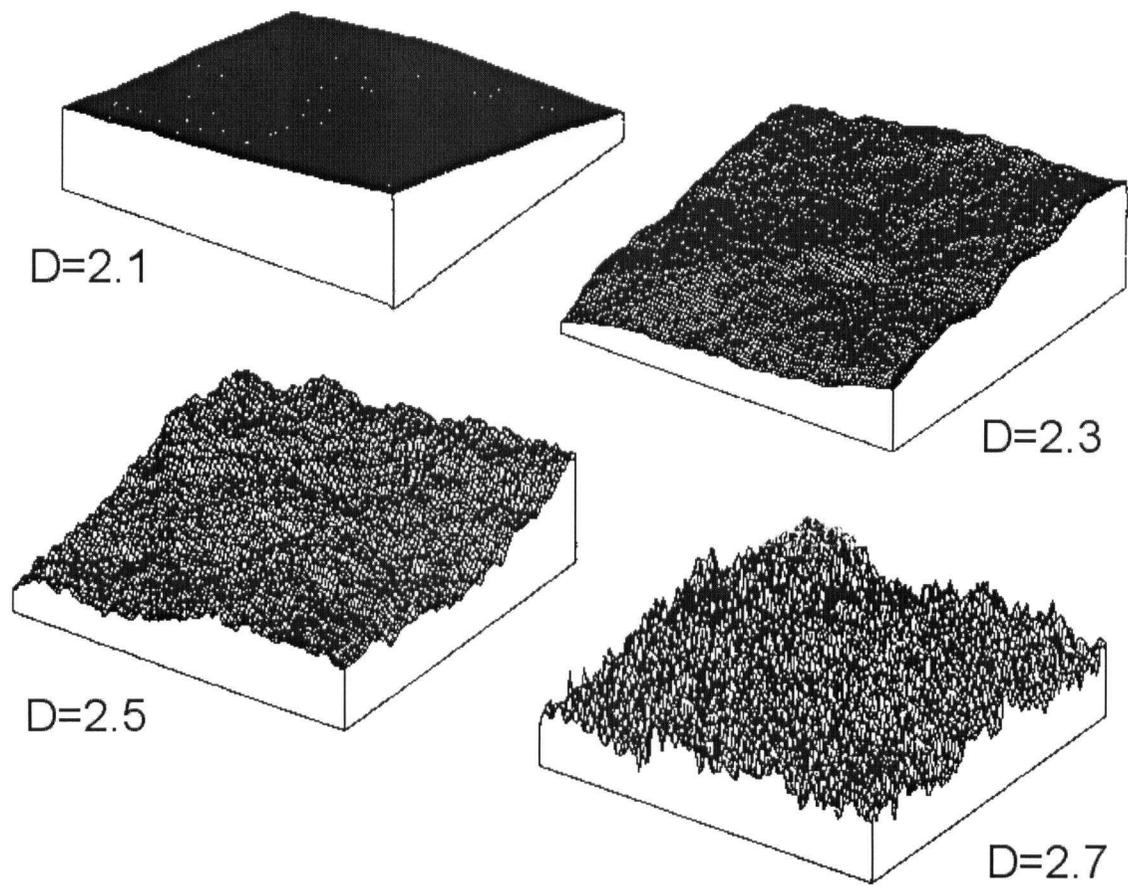


Figure 1.2: Simulated Fractal Landscapes: fractional Brownian surfaces (Klinkenberg and Goodchild, 1993)

Burrough (1981) was among the first geographers to apply fractal concepts to landscapes. His interim assessment was optimistic, but he urged practitioners to refrain from using the fractal concept when it was valid over only a small range of length scales and to seek physical reasons for the appearance of fractal forms. This concurs with the view of Leo Kadanoff (1986), a physicist whose initial reaction was cautiously optimistic. Both Burrough and Kadanoff recognized that fractal geometry is well suited to describing geographical and physical patterns, but can be poor at describing the processes by which these patterns are generated. A similar view was put by Church and Mark (1980), although fractal geometry was not the primary focus of their discussion. A contemporary reading of Church and Mark provokes the same view of fractals as of scale relations: they are a useful start but, through paucity of process, a poor conclusion.

Throughout the 1980s, attempts were made to measure the fractal dimension of geomorphic features. Naturally, most success was found in the prototypes set out by Mandelbrot in 1983:

- Branching river networks (Tarboton *et al.*, 1991; Rodríguez-Iturbe and Rinaldo, 1997).
- River meanders (Snow, 1989; Nikora, 1991; Stølum, 1996).
- Topographic transects and fields (Klinkenberg and Goodchild, 1992).

In general, mixed results were obtained: some were encouraging to supporters of the fractal geometry of nature, others showed results that discounted fractal hypotheses.

If the 1980s were characterized by the measurement of fractal objects, the 1990s were dominated by attempts to link fractal form and physical or geomorphic processes. In the context of river basins, much of this work is summarized by Rodríguez-Iturbe

and Rinaldo (1997), two of the major participants in researching the fractal-process link. The current state of fractal research in geomorphology can be summarized in the following three views:

- The fractal concept is useful in process-response models in so far as simple mono-fractal (or unifractal<sup>5</sup>) scaling results from the underlying physics of some landscape processes;
- Fractals are useful, as above, but landscapes are not simple mono-fractals;
- The fractal dimension is a useful geomorphometric statistic that describes surface roughness, but it cannot be related to the physics of process-geomorphology.

The first view, that landscapes are simple mono-fractals, was prevalent following Mandelbrot's thesis (1983). Turcotte (1992) demonstrated that topographic transects are examples of self-affine fractals (a self-affine fractal is one that scales in different ways in different directions).

Further work has shown that in most cases landscape variability is not this straightforward. Lavallée *et al.* (1993) claim that multifractal concepts are useful in characterizing land-surfaces. A multifractal is a set on a geometric support; it may be formed by a multiplicative cascade, or by sampling a set with a spatially variable fractal dimension (Rodríguez-Iturbe and Rinaldo, 1997). Multifractal statistics are a natural extension of the fractal dimension, wherein higher order moments of the distribution under investigation are studied. A set is multifractal if the relation between moment order and moment variogram-slope is non-linear. Figure 1.3 shows a linear moment-slope relation, indicating mono-fractal scaling; for a multifractal, the moment-slope plot would be non-linear.

---

<sup>5</sup>Evans and McClean (1995) point out that "unifractal" is more consistent etymologically, however "monofractal" is more common in the fractal literature: "mono-fractal" will be used here to accommodate both views.

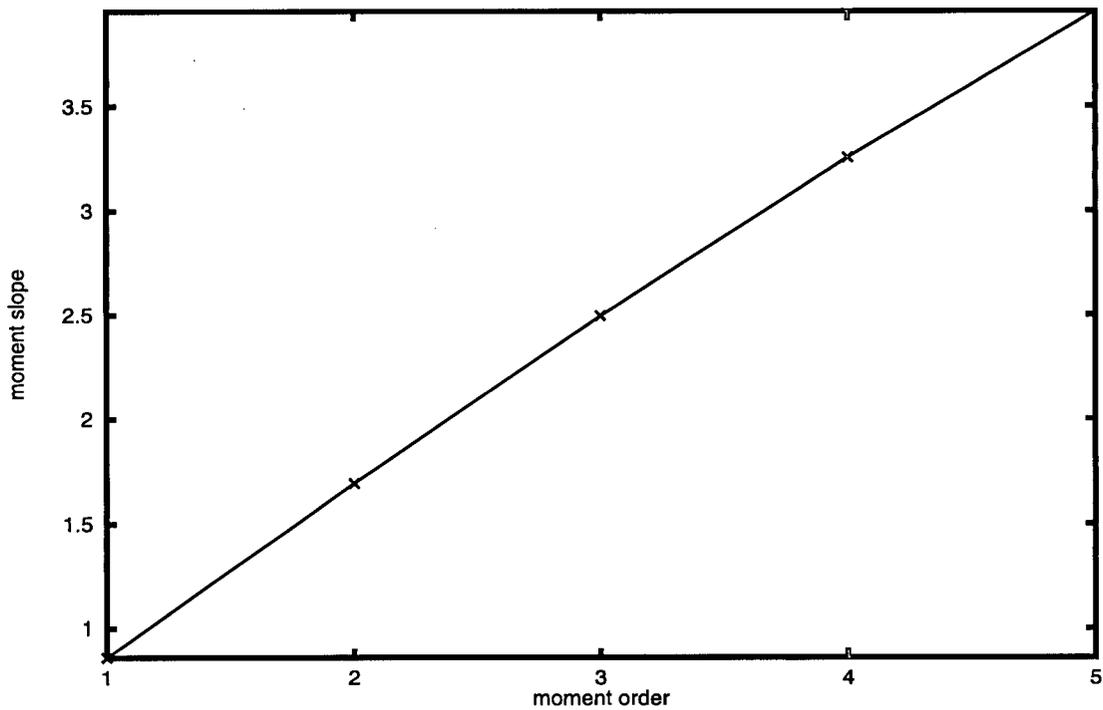
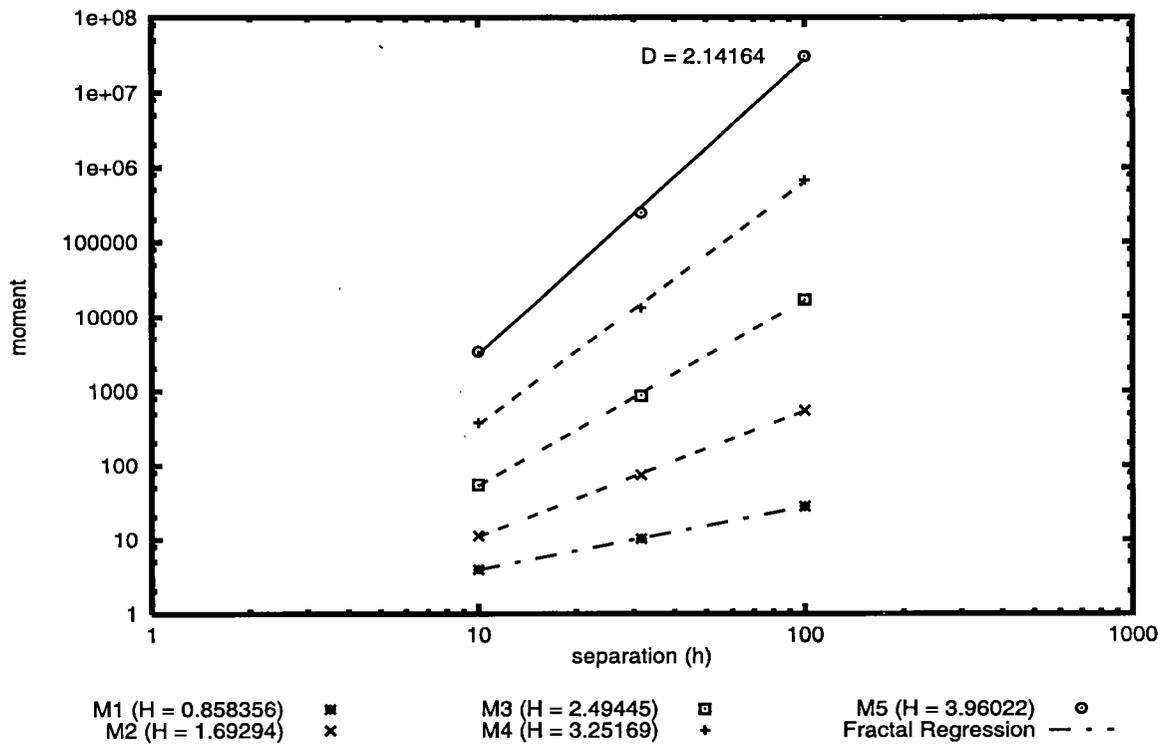


Figure 1.3: Moment statistics of a land-surface (part of Daniels Creek in the Capilano Watershed; see Chapter Four). The lower of the two plots shows a linear moment-order relation, indicating that this particular landscape is a mono-fractal

In parallel to the positive reaction to Mandelbrot's (1983) *Fractal Geometry of Nature* came a more sceptical reaction (e.g., Kadanoff, 1986; Shenker, 1994). The central tenets of this objection are summarized neatly by Avnir *et al.* (1998) in a review of a sample of empirical fractal claims published in the physics journal *Physical Review Letters*. These objections are: (i) empirical fractals cannot continue indefinitely without violating other physical laws; and (ii) most studies of empirical fractals demonstrate scaling over a range of fewer than three orders of magnitude. Avnir *et al.* (1998) suggest that an understanding of these finite-scaling structures may be impaired by their treatment as fractals: they appear to be fractal, but are probably not.

In landscape-scale geomorphology, Burrough (1981, 1993) has introduced the term pseudo-fractal to describe phenomena that appear fractal, and for which there is practical utility in measuring the fractal dimension,  $D$ , but the genetic processes of which are not fractal. In the present context this is appealing: erosive processes remove landscape material within a wide, but inherently limited, range of length scales. It seems impossible for a process that does have a characteristic length scale (or limited range of characteristic length scales) to create a land-surface with no characteristic length scale (or much wider range of spatial variability). However the cumulative effect of many scale-bound processes operating on the same surface may be to create a pseudo-fractal landscape. One goal of the present study is to demonstrate the effect on the fractal dimension of a surface of perturbations caused by large slope failures, and therefore to determine whether or not landscape processes with a wide but finite range of length scales can create the pseudo-fractal landscape that Burrough (1993) suggests.

Veneziano and Iacobellis propose a compelling merger of fractal and pseudo-fractal theories of landscape that avoids the complexities and dissatisfactions of a

multifractal approach. They show that the combination of fluvial and hillslope processes can create complex fractal landscapes across which the fractal dimension is not stationary. They suggest that fractal analyses of topography should segregate between fluvial and hillslope process-domains. Their method will be used here in order to examine the link between the nature and rate of hillslope mass-wasting processes and the fractal dimension of topography in the hillslope process-domains.

### 1.3 Objectives

The first objective of this thesis is to test whether distinct fractal domains can be identified in the landscape. It is hypothesized that if landscape processes can be organized into process-domains, then their impact on landscape morphology ought to be to create areas of homogeneous fractal dimension. In order to test this hypothesis, several digital elevation models are analysed, after having been partitioned into the process domains suggested by Veneziano and Iacobellis (1999).

The second objective of the present work is to examine whether process domains with similar fractal dimensions may be reproduced using a field-calibrated model of landscape evolution incorporating fluvial erosion, diffusive hillslope processes and episodic deep-seated bedrock landslides. These constitute some of the most important geomorphic processes in the areas chosen for empirical study, which are described in Chapter Two.

## Chapter 2

# Surface Processes and Tectonic History in the Queen Charlotte Islands and Southern Coast Mountains of British Columbia

### 2.1 Location

Two areas are used to test the work reported here: parts of the Queen Charlotte Islands (Haida Gwaii), British Columbia, and the Greater Vancouver Regional District (GVRD) Capilano Watershed, within the Southern Coast Mountains of British Columbia. Figure 2.1 gives the regional situation for these two sites, which form part of the Canadian Western Cordillera. Figures 2.2 and 2.3 show details of the location of the study areas.

The purpose of this chapter is to provide tectonic, geological and climatic context in order that modelling and analysis decisions may be justified, and results interpreted. Beginning with the geological background and progressing to a review of surficial processes, the differences and similarities between the two areas are discussed. The importance of tectonic structure and earth history in the development of the drainage network is also considered.

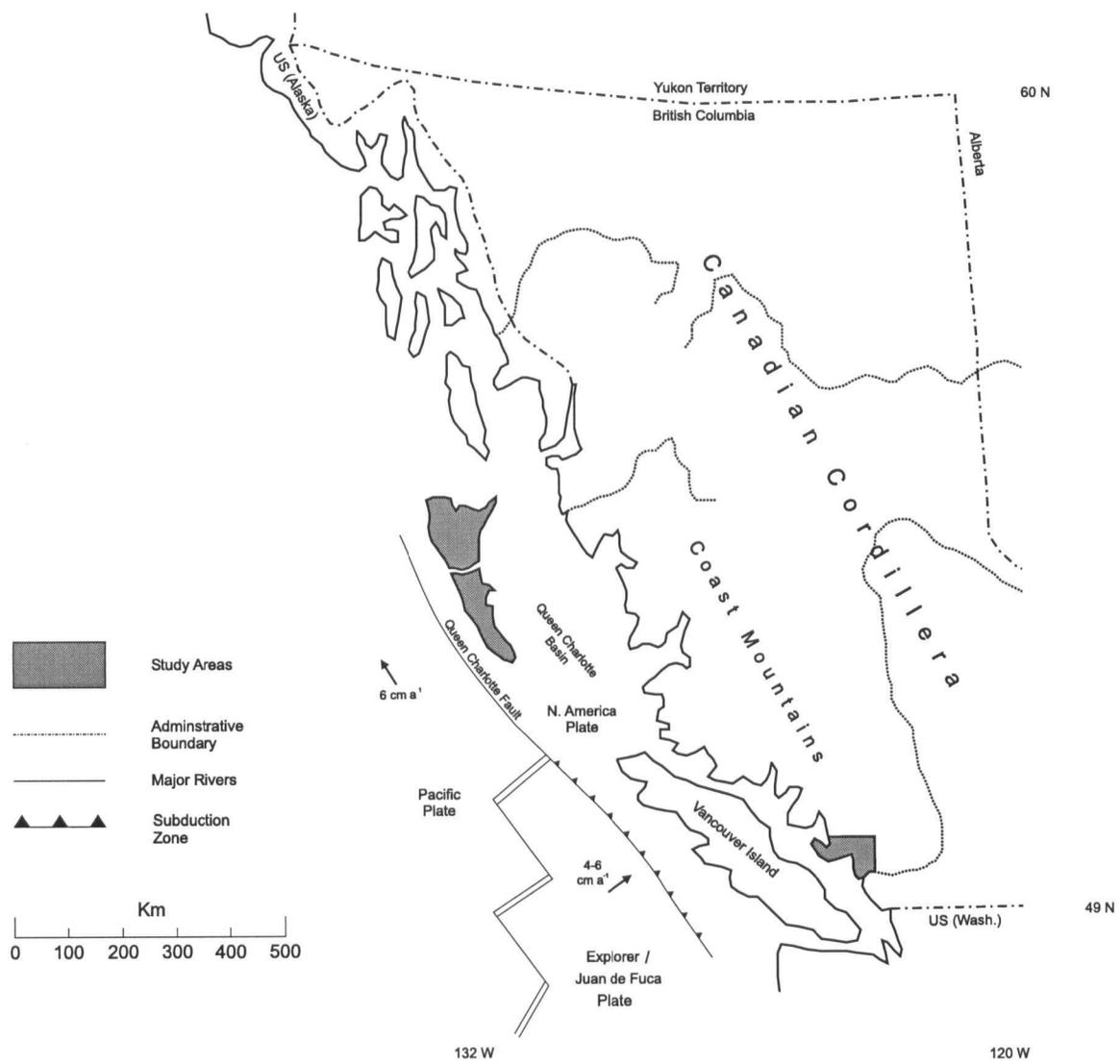


Figure 2.1: Location of the two Study Areas (also showing principal tectonic features, described in this chapter)

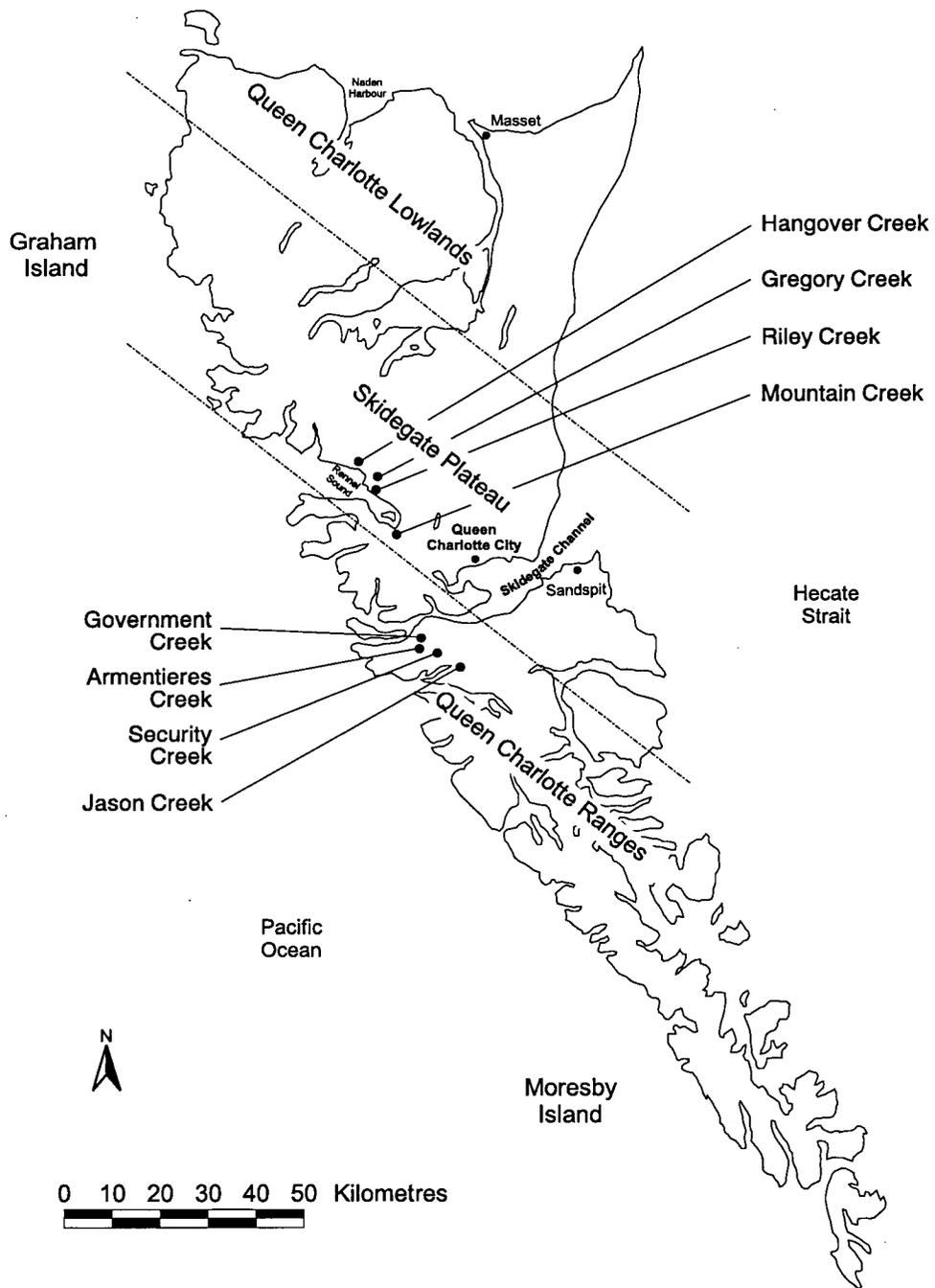


Figure 2.2: Queen Charlotte Islands (study basins are described here and in Chapter Four)

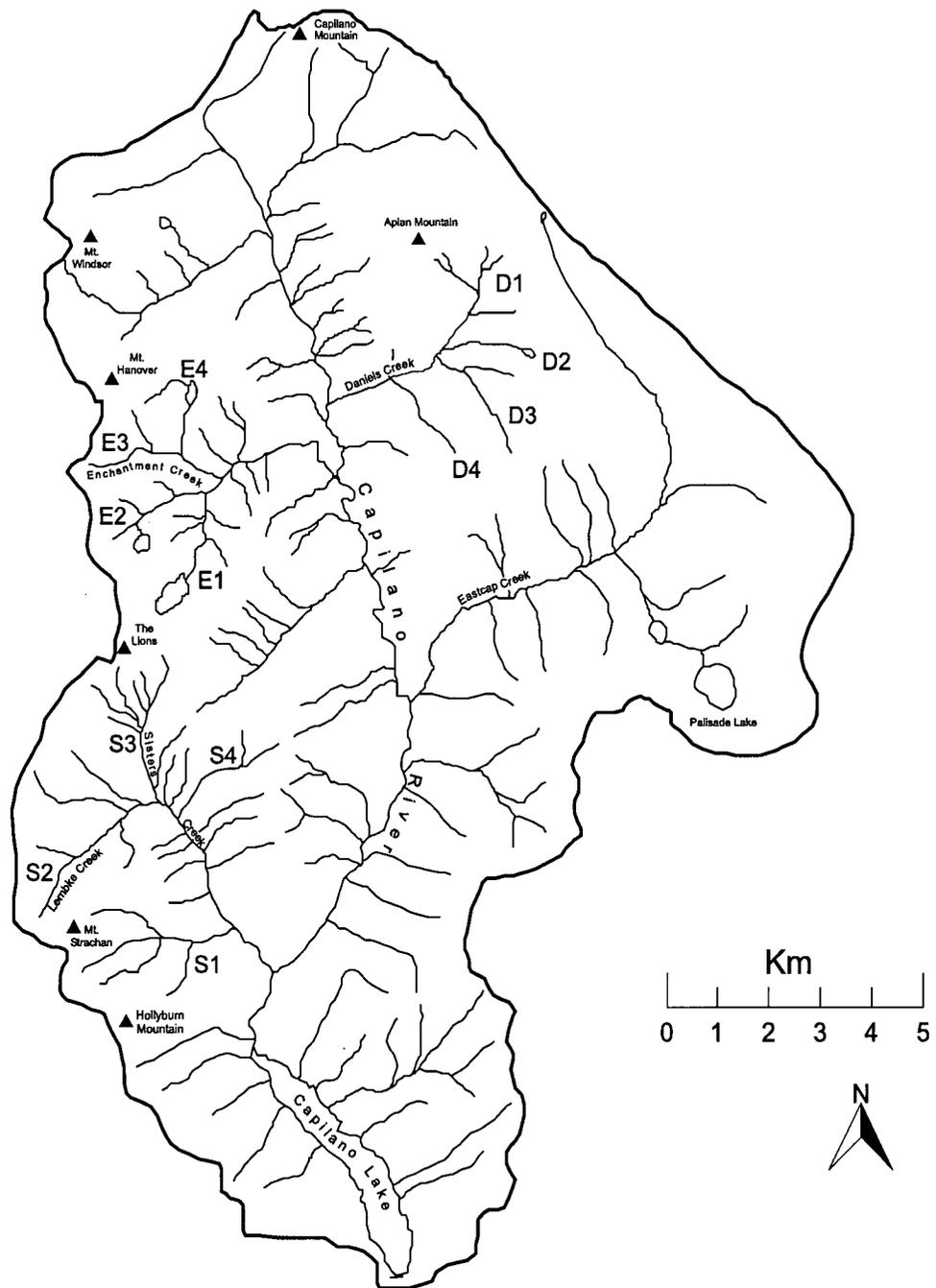


Figure 2.3: GVRD Capilano Watershed (study basins are described here and in Chapter Four)

## 2.2 Tectonic background and Geological History

The two areas lie within the same general tectonic region on the western edge of the Canadian Cordillera, close to the triple junction between the Pacific, Juan de Fuca/Explorer, and North America plates, which lies approximately 100 km west of Vancouver Island (Parrish, 1983). The tectonic regime is influenced by the relative rates of motion of these plates, given in Figure 2.1. South of the triple junction, subduction occurs offshore, where the Juan de Fuca/Explorer plate passes under the North America plate (Figure 2.1). North of the triple junction, is the Queen Charlotte Fault, which separates the Pacific and North America plates. This transform fault is the most seismically active in Canada, a fact that has profound consequences for slope stability on the Queen Charlotte Islands.

The most recent period of mountain building in the Coast Range began 10 Ma BP, with a maximum average rate on the order  $0.5 \text{ mm a}^{-1}$  Parrish (1983). More recent uplift rates of up to  $2 \text{ mm a}^{-1}$  have been inferred from tidal records. In some heavily glaciated areas this uplift may be isostatic in origin, whereas in others the movements are mainly tectonic (Clague, 1989). In the following sections, these general rates will be supplemented with information about the nuances of each site.

### 2.2.1 Tectonics and Topography

The long-term interactions between tectonics and topography have been the subject of much recent work (Molnar and England, 1990; Beaumont *et al.*, 2000). Uplift increases the amount of potential energy available for erosion by surface processes. This is important because, over long timescales, lowering of the land-surface (resulting from denudation by surface processes) is accompanied by a considerable removal of landscape mass. For normal crustal rock densities, this leads to further (isostatic) uplift amounting to approximately four-fifths of the original lowering.

Over short timescales, the relation between tectonic processes and landscape processes is also relevant to studies of the observed topography. The influence of linear tectonic structures (lineaments, including faults and thrusts) on large landslides has been the subject of several studies in the Coast Mountains. Leir (1995) and English (1996) have shown that the locations of structure-forming landslides are strongly influenced by the location of lineaments. In the present landscape, deep-seated landslides occur with a lower rate than is likely either after glaciation (Ryder, 1981) or in the very early history of the Cordilleran orogen (cf. Hovius *et al.*, 1998). In the current study area, the influence of early slope failures on the present drainage network may have been obfuscated by the later occurrence of several widespread glaciations, which provide an immediate explanation of many important features of the contemporary landscape. Nevertheless, the pre-glacial topography in regions such as this may have been strongly influenced by widespread landsliding associated with orogenic activity. If this is true, landslides would have provided significant depressions in which the first cirque glaciers of subsequent glaciations may have formed. Early controls on the development of the drainage network are beyond the temporal scope of this study, however the influence of large contemporary slope failures on the drainage network is evident in the current landscape.

### 2.3 Queen Charlotte Islands

The Queen Charlotte Islands (Haida Gwaii), lie approximately 100 km off the west coast of British Columbia. They are part of the insular tectonic belt of the Canadian Cordillera and one of the most seismically active regions in Canada, being close to the Queen Charlotte Fault. The islands have been uplifted by a maximum of 1 km in the last 10 Ma, according to the fission-track observations of Parrish (1983).

### **2.3.1 Bedrock Geology**

The Queen Charlotte Islands comprise three principal physiographic provinces: the Queen Charlotte Ranges, the Lowlands, and the Skidegate Plateau. Details of the bedrock geology are given in Sutherland-Brown (1968): the majority of the study area is underlain by early Tertiary lavas, which are gently bedded or flat. Sutherland-Brown (1968) divides the volcanics into several formations. The "soft" volcanics of the Masset and Yakoun formations are common: they have highly developed joints (especially in the columnar basalts), which render the rocks highly permeable and therefore susceptible to extensive sub-surface weathering. As a consequence, the area is prone both to deep-seated and superficial slides; indeed, the majority of recent rock slides have occurred in this formation (Alley and Thompson, 1978). Another important formation for slope failures is the "hard" volcanic Karmutsen group, found along the Skidegate Channel between Graham Island in the north and Moresby Island in the south. A number of recently intruded granitic plutons are of local significance, being responsible for the relatively high incidence of slope failure near Pivot Mountain and in Riley and Bonanza Creeks. Most notably, there is an area of sedimentaries in the north (Kunga and Longarm formations); it is not well-weathered, but is deeply gullied owing to a high degree of jointing. Weathering rates are higher for the Longarm sedimentaries, so some surficial slides have been observed (Alley and Thompson, 1978).

### **2.3.2 Glacial Legacy**

The Fraser Glaciation had an important effect on the landscapes of the area. Alley and Thompson (1978) note that at the peak of this most recent Cordilleran glacial episode there was an ice cap approximately 900 m thick over the area, which coalesced with the main Cordilleran ice sheet. The retreat of this ice cap, which occurred in line

with the retreat of the mainland ice sheet (between 14 and 12 ka BP), left a veneer of till masking the surface and has resulted in at least 18 m of isostatic rebound since 5 ka BP (Sutherland-Brown, 1968).

During deglaciation, a period of low sea level (from 15 ka to 10 ka BP) was followed by a rise until 8 ka BP leaving some shorelines 15 m higher than at present Clague (1989). Clague (1989) suggests that ice loads on the Queen Charlotte Islands were insufficient to cause isostatic depression. In addition, the eastward migration of mantle material following the decay of the mainland icesheet provides a further explanation for the general rise in sea-level on the Queen Charlotte Islands during the Holocene.

### **2.3.3 Contemporary Processes**

The contemporary climate is one of cool summers and mild winters. Annual precipitation at lowland settlements is generally 1,000–1,250 mm, but some western slopes receive up to 7,600 mm per year. This falls mainly as rain, with snow in winter at altitudes above 600 m. A further meteorological feature that is worthy of note is the prevalence of violent winds, gusting in excess of 190 km/h, which frequently lead to tree-throw.

Seismic activity has direct consequences for slope stability. Alley and Thompson (1978) comment on 1,268 recorded earthquakes between the years 1899 and 1974, with 31 between Richter magnitudes 5 and 7; 7 between magnitudes 7 and 8; and 1 of magnitude greater than 8 (in August 1949).

The combination of steep slopes, highly jointed rock, a failure-prone glacial veneer, high levels of intense winter storm precipitation, violent gusts of wind, and frequent seismic shocks results in a dynamic landscape in which slope failures are

frequent and sediment yields are high. Rates of sediment flux due to landsliding were measured by Rood (1984, 1990) and Gimbarzevsky (1988). A summary of the rates in each formation is given in Table 2.1.

Formation	Type	Average Yield $m^3 ha^{-1} a^{-1}$
Masset	soft volcanics	1.5
Karmutsen	hard volcanics	2.0
Haida	sedimentaries	2.0
PTP	granites	1.3

Table 2.1: Sediment Flux due to Landsliding on Queen Charlotte Islands (data from Rood 1984)

## 2.4 GVRD Capilano Watershed

The Capilano Watershed, in North Vancouver, is part of the Pacific Ranges of the Coast Mountains. The area is characterized by steep slopes (commonly greater than 35 degrees) and high relief (between 900 and 1,300 m over 2 or 3 km), which lead to active hillslope processes and high sediment yields. Resulting from the needs of watershed planners, there is a well-presented record of these processes available for use in the present study, including gully processes, shallow landsliding and (to a lesser degree) larger rock slides. The following sections are based on this inventory (referred to as GVRD, 1999), and on the earlier synthesis of (Ryder, 1981).

### 2.4.1 Bedrock Geology

Bedrock in the area comprises intrusive igneous granodiorite, quartz diorite, diorite and lesser amounts of gabbro and migmatite, with great intact strength and widely-spaced joints. There are small areas of volcanic rocks, which are similar but more

highly jointed than the intrusives. Bedrock is exposed at mid and upper elevations; at lower elevations, there is a mantle of glacial drift and colluvial deposits.

### **2.4.2 Tectonic Uplift**

Uplift history in the Capilano watershed is similar to that of the Queen Charlotte Islands, and was the subject of the same regional study by Parrish (1983). However, in the last 10 Ma, the GVRD Watershed has seen between 2 and 3 km of uplift (cf. 1 km on the Queen Charlotte Islands). Ryder (1981) highlights the importance of a pause in uplift late in the Tertiary period, which allowed the development of an erosion surface that was further dissected upon subsequent uplift (but is currently visible in some upland parts).

### **2.4.3 Glacial Legacy**

The main Cordilleran ice sheet covered the GVRD Water Supply Area, which has been subject to a larger number of Pleistocene glacial repetitions than documented on the Queen Charlotte Islands (where the ice cover was separate from the mainland). Evidence of three major glaciations has been found in the Lower Mainland, although the most recent (the Fraser Glaciation) dominates the landscape.

The Fraser Glaciation in the study area is described by Lian and Hickin (1993). It began roughly 30 ka BP with slow development of cirque glaciers in high, north-facing depressions, expanding until about 20 ka BP, when large valley glaciers had developed (Coquitlam Stade). Between 18 and 16 ka BP, the Port Moody Interstade saw a recession of the ice and reforestation of lowlands. Subsequently, the ice re-advanced until the glacial maximum at 14.5 ka BP, when a 1.5 km thick ice-tongue extended well into Washington State, USA. Ice retreated rapidly and it is likely that

it had melted from even the higher parts of the Capilano watershed before 10 ka BP.

Whereas sea levels have risen since deglaciation on the Queen Charlotte Islands, on the mainland a period of isostatic rebound immediately followed the retreat of ice, leading to a marine regression in spite of global eustasy. In the first 2–4 ka since the retreat of ice, isostatic rebound of formerly glaciated areas was larger than the coeval eustatic rise, leading to a reduction of shoreline elevation by 100–200 m. However, in the last 8–10 ka shoreline elevation has risen again by about 20 m to its present level, probably as a result of global eustatic rise in sea level and forebulge collapse (Clague, 1989).

Erosion rates in the watershed prior to the revegetation of slopes were considerably higher than at present, with a much higher incidence of both deep-seated rock slides and shallow slides in the veneer of glacial material (Ryder, 1981; GVRD, 1999).

#### **2.4.4 Contemporary Processes**

The present climate is characterized by large frontal depressions that track successively across the area between October and March, and summer high pressure that is occasionally interrupted by small intense storms that may produce extreme flooding.

These conditions, combined with contemporary tectonic uplift, steep slopes and a residual veneer of glacial deposits result in high rates of hillslope failure, which are documented by GVRD (1999), Savigny (1991) and English (1996). Campbell (1999) recently determined a sediment yield due to landsliding of  $168 \text{ m}^3 \text{ km}^{-2} \text{ a}^{-1}$  ( $= 1.7 \text{ m}^3 \text{ ha}^{-1} \text{ a}^{-1}$ , similar to the values reported for the Queen Charlotte Islands) for the nearby Lynn Valley.

## 2.5 Summary

The purpose of this chapter is to present the geological and geographical characteristics of the study areas, giving some published estimates of background rates and summary process statistics. In particular, it highlights the findings of Ryder (1981), Rood (1984, 1990), English (1996) and GVRD (1999), that contemporary processes and forms are strongly influenced by the tectonic structures in the area, and the glacial overprint. In Chapter Three, the rates and magnitude-frequency characteristics will be examined in more detail, so that their influence on landscape form may be assessed.

## Chapter 3

### Sediment Flux due to Episodic Events

#### 3.1 Process rates

Estimates of sediment transport by episodic processes presented in the previous chapter indicate that landslides and debris flows play an important role in the long-term denudation of the GVRD Water Supply area and the Queen Charlotte Islands. Quantification of sediment flux due to episodic processes is an important step in the construction of testable models of long-term landform evolution (Martin, 1998, 2000), and has important consequences for the resulting landscape form. In this chapter, the record of slope-failures in the two study areas will be examined in order to determine whether they follow a power-law magnitude-frequency distribution, and to discuss how they can be incorporated into a landscape evolution model.

#### 3.2 Sources of Data

The statistical analyses presented in this chapter are based on three previously published landslide datasets: (i) a set of 1,337 failures mapped by Rood (1984, 1990) for parts of the Queen Charlotte Islands; (ii) a set of 865 landslides collected as part of

the GVRD Ecological Inventory Program (GVRD, 1999); and (iii) a set of 49 major rock slides collected by Savigny (1991).

### 3.3 Surficial Slides: magnitude-frequency

#### 3.3.1 Results

The results presented here are from the GVRD and Queen Charlotte Islands study areas. Figure 3.1 shows a magnitude-frequency distribution for 1,337 failures on the Queen Charlotte Islands (the dashed line). A similar distribution for the GVRD landslide inventory is also shown in Figure 3.1 (the dot-dashed line), although in this latter case there is little preserved evidence of a power law. In each case the slide volumes extend over a range of two to three orders of magnitude. For the Queen Charlotte Islands landslides, the power-law in Equation 3.1 (the solid line on Figure 3.1) was fitted to the most robust central part of the curve (between 400 and 7,000 m<sup>3</sup>).

$$\text{Frequency} = 0.8 * \text{Magnitude}^{-0.63} \quad (3.1)$$

The exponent here is negative, indicating that frequency of events declines with their magnitude; it is also less than one, which shows that large events are more important than the cumulative effect of small events (Hovius *et al.*, 1997). This is of particular interest because it implies that small events may be modelled with diffusion provided that large events are considered explicitly. There are two interesting features at each end of the distribution: (i) a relative lack of small events (< 400 m<sup>3</sup>), probably a result of the limited sampling resolution and rapid revegetation of landslide remains, but possibly a dynamical feature of the landsliding process; and (ii) a break of slope for large events (> 7,000 m<sup>3</sup>), which may indicate a change in controlling conditions or another sampling limitation.

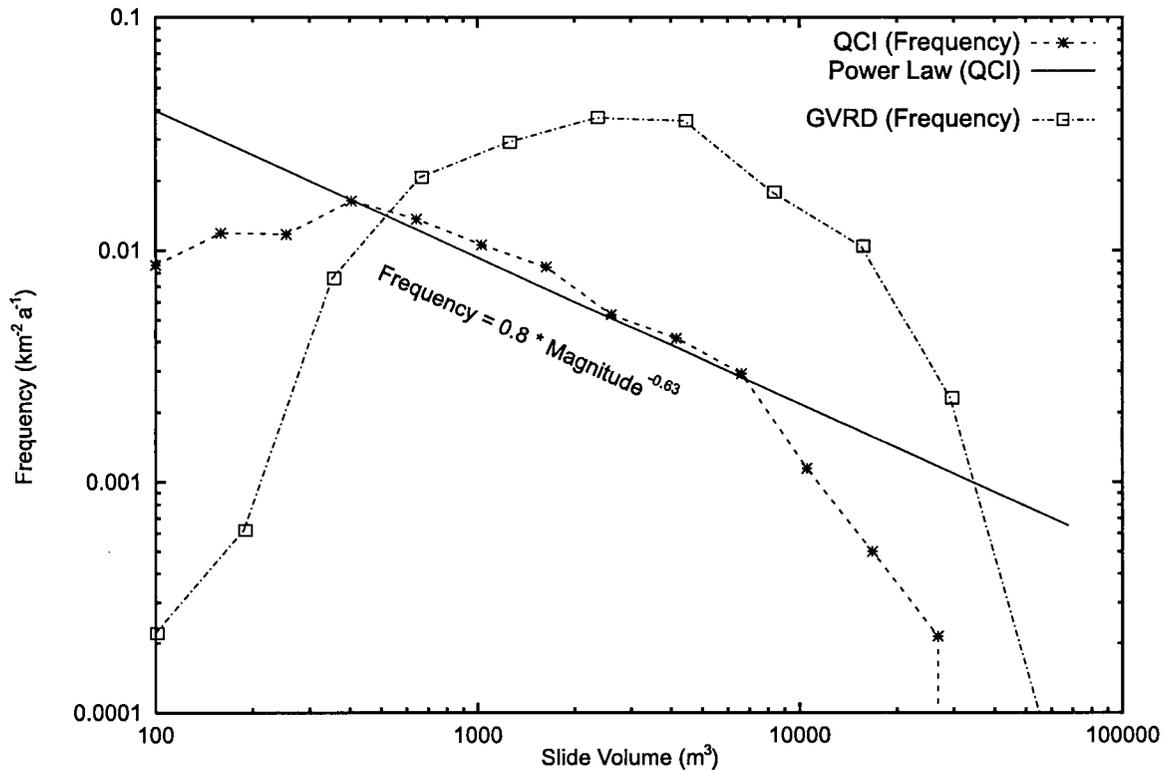


Figure 3.1: Queen Charlotte Islands: Landslide Magnitude Frequency

### 3.3.2 Power-law Scaling and Self-organized Criticality

Several hypotheses have been offered to account for power-law scaling in natural phenomena. These include the dynamical state of self-organized criticality, and the possibility that a power-law may arise from imperfect sampling of rare events. These competing explanations are elaborated below.

Power-law scaling may be a natural feature of hillslope instability (Hovius *et al.*, 1997; Pelletier *et al.*, 1997; Turcotte, 1999). Failures resulting from the exceedence of critical thresholds commonly have power-law distributed magnitude distributions; self-organized criticality has been offered as an explanation for this by Bak *et al.* (1987, 1988).

Self-organized criticality (SOC) is a feature of systems maintained far from equilibrium that organize themselves through internal feedbacks to a state in which a system parameter fluctuates with power-law distributed adjustments around a marginally-stable critical value (Bak *et al.*, 1987; Turcotte, 1992). In the present context, slope failures may be the power-law distributed adjustments to which Bak *et al.* refer. This analogy follows naturally from their definitive example of a growing pile of dry sand. In this demonstration, successive grains of sand are added to a circular table. Initially the sand grains form a pile that is stable because its angle is below the angle of repose for sand. As grains continually fall the material in the pile reaches the angle for failure. The claim of Bak *et al.* is that further grains will not simply roll down the stable surface of the pile. Instead, owing to the configuration of the grains, the pile will grow beyond the critical angle by some amount until an avalanche (sand-slide) occurs. Bak *et al.* present a variety of experimental evidence to show that the magnitude-frequency distribution of the resulting avalanches follows a power-law.

The direct analogy of the sandpile to a layer of surficial hillslope material is initially appealing, although Mehta (1992) shows that there are significant differences. Recently, a more physically-based justification of the power-law distribution of slide magnitudes has been presented by Hergarten and Neugebauer (1999), who show that power-law distributed landslides result from a simple model of threshold instability on an inclined plane with one Dirichlet boundary (i.e., constant fluvial downcutting along one side). These theoretical conclusions concur with the empirical findings and explanations of Hovius *et al.* (1997) and Pelletier *et al.* (1997).

An alternative explanation of the proposed power-law is obtained through a consideration of the natural timescales over which contingent hillslope stability criteria apply. In most cases, slope failure is a result of interactions between slope loading by soils and vegetation, strengthening through the development of a root network and

strength-reducing factors (especially pore water pressure in soils). Since these each have different natural timescales (pore water pressure typically changes more rapidly than vegetation changes can occur), it is likely that observations over a discrete time-interval will show a large number of small events and a small number of large events; possibly resulting in a finite-scaling power-law magnitude frequency distribution of slide volumes. Nevertheless, there are some differences between a theoretical power-law and the distributions obtained from field and air photo data, which relate to sampling biases discussed below.

### **3.3.3 Power-law Scaling and Sampling Bias**

The nature of rare and episodic events is that the largest events are usually the most rare, and the smallest events are the most common. This introduces a natural bias to any data collected on such events: all observations must be collected over a limited time period and a restricted area, therefore a large number of small events and a small number of large events will be recorded.

Further temporal or spatial sampling biases may result if the temporal range of observations is too short, in which case a survey may underestimate the number of large events and therefore overestimate the importance of small events. Similarly, if the sample area is too small, the survey may miss large, rare slides (Figure 3.2).

### **3.3.4 Finite Scaling Effects**

The kink in the Queen Charlottes slide distribution, at  $400 \text{ m}^3$ , may be another sampling effect. This represents a relative paucity of events with small volumes (roughly a cube of 5–7 m on each side). One explanation of this feature is that a proportion of these events is missed in the air photo analysis either because the events are too

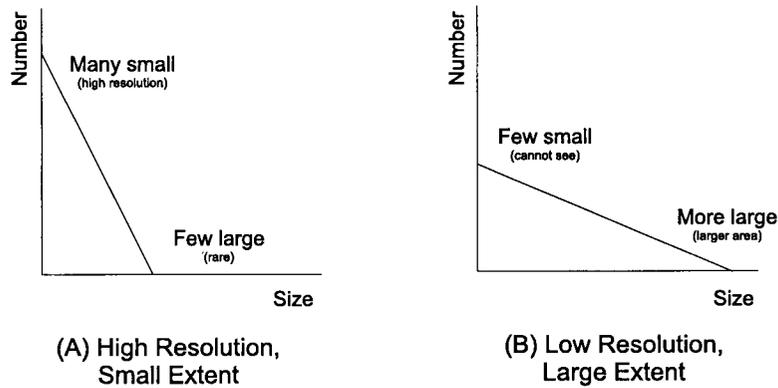


Figure 3.2: Extreme Event Distribution

small to be observed, or because small events are more rapidly and easily obscured by revegetation.

Smith *et al.* (1986) have shown that the persistence of slide evidence is a logarithmic function of time, suggesting that the probability of observing a slide can be written as a function of slide size:

$$Pr(obs) = \exp[(\log P(ref) * cutoff)/size] \quad (3.2)$$

where cutoff is the size (in  $m^3$ ) of a reference slide visible with probability  $P(ref)$ . An example is plotted in Fig. 3.3, where  $cutoff = 2,500$  and  $P(ref) = 0.5$ .

If this resolution function is convolved with an artificially generated set of slide magnitudes chosen to obey the expected power-law the observed landslide distribution also shows a kink at low magnitudes similar to that found in the real data set (compare Fig. 3.4 with Fig.3.1).

Departures from power-law distribution may be due to sampling biases. However the observed kink at low magnitudes may be a genuine feature of the mechanics of slope failure: it may result from the need to accumulate a critical amount of sediment in order that its weight may overcome frictional resistance, consequently fewer

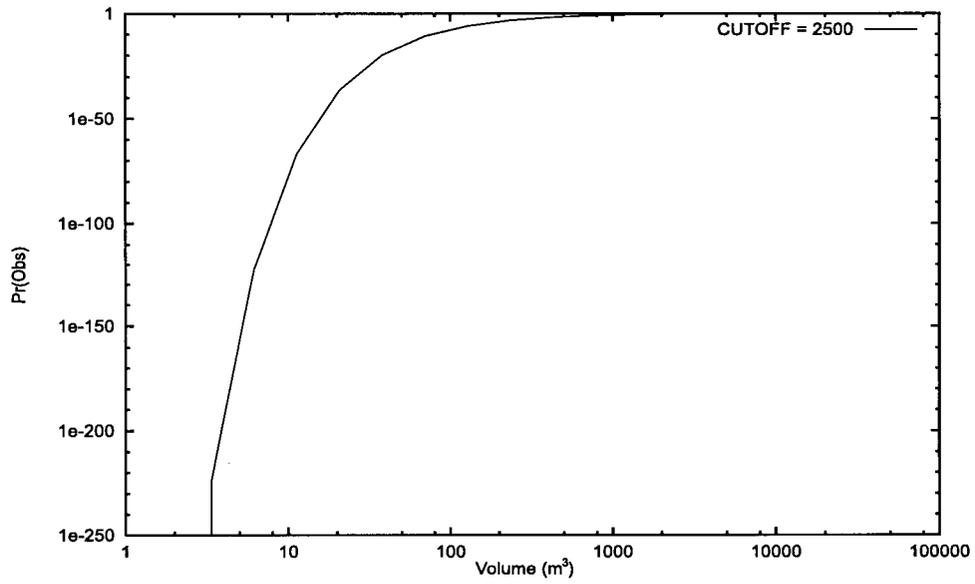


Figure 3.3: Resolution Function

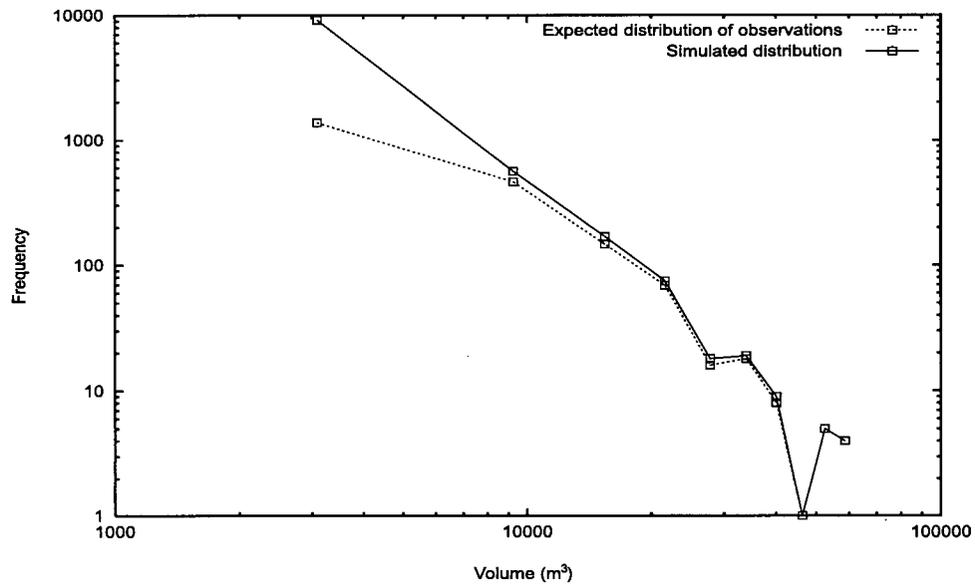


Figure 3.4: Landslide Magnitude Frequency modified by Resolution Function

slides occur with volume lower than a critical amount which may be represented by the observed kink.

To decide between these two hypotheses is beyond the scope of the present work, however it is anticipated that a critical test between the two could be performed in an experimental study of failure of a naturally occurring surficial material: if failures occurred on all scales down to the dimensions of individual grains then the kink observed in the distribution of real landslides is likely to be an artifact; if there is a measurable lower limit to the size of failure that can occur in the experimental case, then confidence increases that finite scaling is an inherent feature of slope failure in natural materials.

### **3.3.5 GVRD Slides**

The magnitude-frequency distribution of the GVRD dataset (the dot-dashed line in Figure 3.1) has a strikingly different shape to the Queen Charlotte Islands dataset. The curve has no linear part, so these data do not follow a power-law distribution. Instead they appear to have a characteristic scale. Two explanations for this result are possible.

The first explanation is that a different resolution function would be capable of creating such an extreme lower-end kink to reproduce the curve shown from an original power-law. The combination of an enhanced lower-end kink and the relative lack of large events (exacerbated by the finite sampling space scale and time interval) leads to the hump observed for the GVRD data. This hypothesis implies that these landslides occur with a power law distribution over a wide range of scales, but it is impracticable to make observations with sufficient resolution, over a wide enough or long enough time, to see the underlying power-law.

The second explanation is that the underlying data are in fact exponentially distributed. This implies that there is a characteristic scale at which landslides occur. Distinction between the two theories is important in the study of the dynamics of mass-wasting. It is suggested that the most fruitful approach would combine experiment (as above) and theory, to determine a suitable distribution for episodic events, which is consistent with their known mechanisms.

### 3.3.6 Implications

The practical implications of surveys of the magnitude-frequency distribution of slope failure are readily apparent in the quantification of risk to humans, their possessions and the structures placed close to landslide hazards. It is worth note that the approach taken here provides probabilistic predictions of landslide statistics, rather than deterministic predictions of the spatial and temporal incidence of the events in question. It is therefore of value in the evaluation of generalized risk, and has applications in actuarial risk analysis, and in the optimal design of risk-mitigating engineering structures (for a recent example in the same area, see Hungr *et al.*, 1999).

The demonstration of a power-law landslide magnitude-frequency distribution also has several theoretical implications: (i) the statistics can be transformed for incorporation into long-term landscape evolution models (see Chapter 5), and (ii) power-law perturbations to hillslope areas may lead to an increase in the fractal dimension of their topography. The second of these points forms the basis of the subsequent chapters of this thesis. In what remains of the current chapter, the landslide statistics are transformed into a non-linear slope stability relation and the influence of very large bedrock landslides is assessed.

## 3.4 Slope and Volumetric Transport Rate

### 3.4.1 Approach

The relation between slope and sediment flux is important because it reveals slope-stability thresholds governed by the dynamics of slope failure. In addition, a constitutive relation (Eq. 3.3) between slope ( $\partial h/\partial x_i$ ) and sediment flux ( $\mathbf{q}$ ) forms an integral part of the approach to long-term landform evolution taken in Chapter 5.

$$\mathbf{q} = f(\partial h/\partial x_i) \quad (3.3)$$

### 3.4.2 Parameterization

In order to predict sediment flux from slope, a suitable function,  $f$ , must be found, and parameters must be fit using the observed landslide data. Martin (1998, 2000) has suggested that an hyperbolic tangent function is appropriate:

$$\mathbf{q} = \frac{K_1}{2} \left[ 1 + \tanh \left( \frac{|(\partial h/\partial x_i)| - h_{x0}}{h_{scale}} \right) \right] \left( \frac{\partial h}{\partial x_i} \right) \quad (3.4)$$

Martin (1998) parameterized Equation 3.4 for the same Queen Charlotte Islands landslides described here, with  $K_1 = 0.43$ ,  $h_{x0} = 1.06$ ,  $h_{scale} = 0.19$ .

The procedure adopted by Martin (2000) for the development of a slope-transport rate for shallow landslides and debris flows on the Queen Charlotte Islands was repeated for the GVRD landslide inventory. Landslide width, length, thickness and initiation slope were reported in the GVRD Ecological Inventory (1999), with measurements made from air photos and calibrated using field data. Volumes were computed using a relation presented in the GVRD analysis, and the slide population was divided into ten slope-classes as shown in Table 3.1. The area falling into each slope class was

estimated from a digital elevation model of the area (also supplied by GVRD). The temporal range of the observed data was estimated as 30 years, using dendrochronology of woody debris found within a sample of the deposits (GVRD, 1999).

Slope Class $m/m$	Yield $m^3 m^{-2} a^{-1}$
0.00–0.25	0.004
0.25–0.50	0.011
0.50–0.75	0.035
0.75–1.00	0.068
1.00–1.25	0.113
1.25–1.50	0.112
1.50–1.75	0.060
1.75–2.00	0.028
2.00–2.25	0.011
2.25–2.50	0.000

Table 3.1: Slope-Volume Classes for GVRD Landslides

A function of the form in Equation 3.4 was fit by eye to the data in Table 3.1 to obtain initial “seed” values for a non-linear least-squares curve fitting routine `nlm` in the statistical package R, which resulted in the relation shown in Figure 3.5, with  $r^2 = 0.86$  (The fitted parameters are shown in Table 3.2). This relation is valid for slopes with gradients between zero and approximately 1.5 (i.e., 60 degrees). On steeper slopes, surficial failures of the type analysed here give way to bedrock-controlled failure mechanisms, such as deep seated rock sliding.

Parameter	Queen Charlotte Islands Martin (1998)	GVRD this study
$K_1$	0.43	0.12
$h_{x0}$	1.06	0.81
$h_{scale}$	0.19	0.32

Table 3.2: Parameters for Non-Linear Slope-Volume Relation

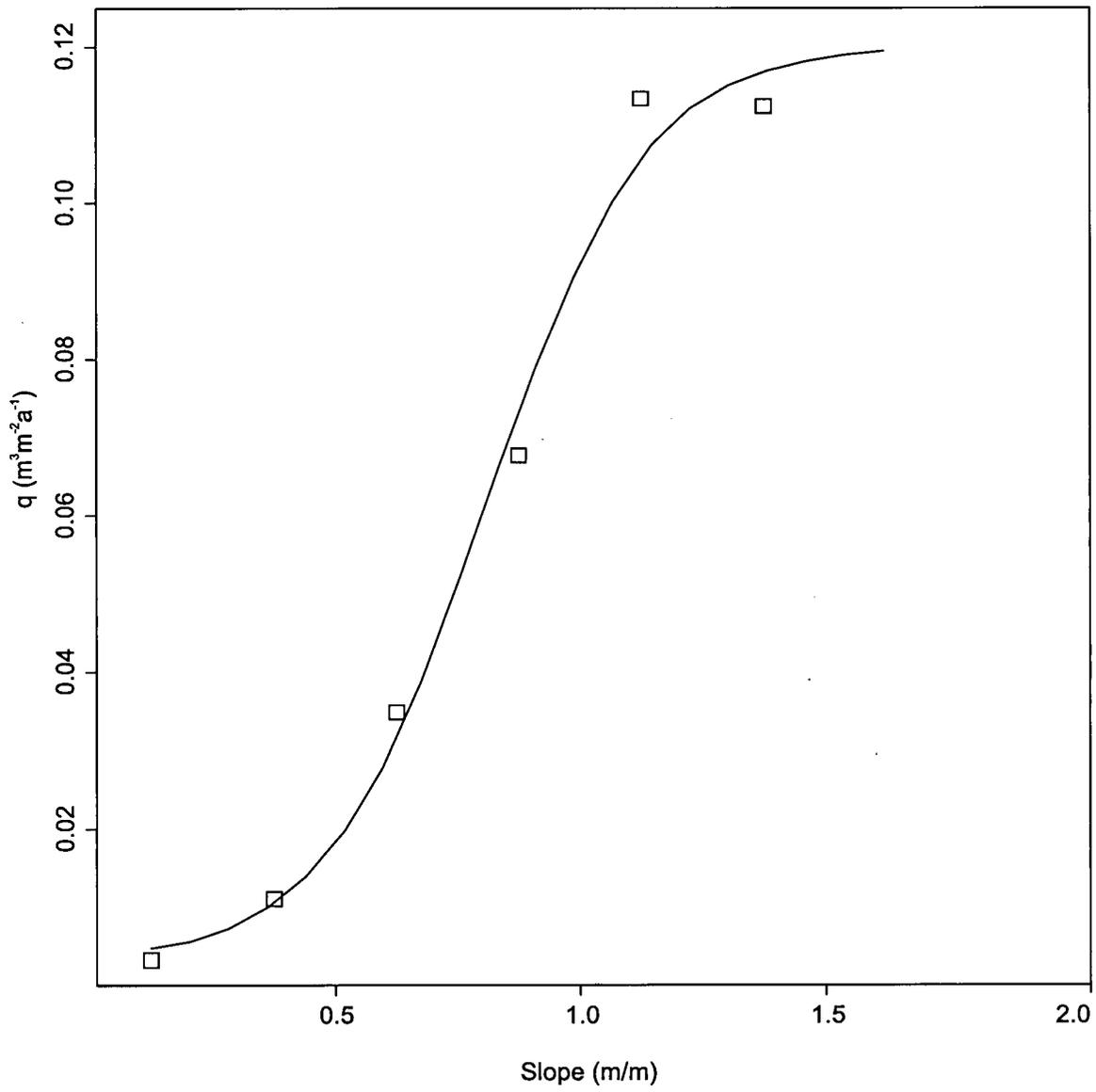


Figure 3.5: GVRD Slope-Volume Relation

### 3.4.3 Discussion

The results reported in the previous section provide strong support for the general applicability of the non-linear transport function proposed by Martin (1998, 2000). They form the basis of the constitutive relation between slope and sediment flux used in the landscape model of Chapter 5. Other non-linear functions have subsequently been proposed (Tucker and Bras, 1998; Roering *et al.*, 1999), however these are a step function and exponential function respectively. The advantage of the hyperbolic tangent, proposed by Martin (1998, 2000) and subsequently used here, is that it correctly identifies the maximum slope on which (shallow) landsliding is important. Beyond this value of slope, stability is governed by factors other than the internal friction of surficial material.

## 3.5 Rock Slides

Large rock slides, which remove material on scales comparable to local relief, are important features in the evolution of landscapes in this area (Savigny, 1991; English, 1996; Hungr *et al.*, 1999). The morphological effect of a rock slide may persist if it is large compared to the rate of its subsequent obliteration (by weathering and erosion). If rock slides occur over a wide range of scales their effect may be to increase the fractal dimension of hillslopes, because they will impart to a hillslope topographic variation over a range of scales. It is appropriate to consider large rock-slope failures as an extraneous shot noise in the present work, since none of their controlling factors besides slope is codified within the landscape evolution model. In this section, the statistics of a regionally-significant rock slide inventory in the region are examined, in order to provide a basis upon which to compare model-generated landslide statistics with those observed in nature.

### 3.5.1 Rock Slide Inventory

One of the largest inventories of rock slides in the region was made by Savigny (1991), who catalogued 49 rock slides in the Fraser Canyon and vicinity. This is the best record of very large slope failures close to the areas in which the present work is based. Savigny (1991) produced a map of slide areas based on the interpretation of 1:50,000 black and white aerial photography, which was digitized to find the slides areas. The magnitude-frequency statistics of this dataset are shown in Figure 3.6.

The record of large landslides extends over more than two orders of magnitude ( $0.2 - 35 \text{ Mm}^3$ ). Their cumulative distribution takes the general form of a power-law, which is similar to that found by Hungr *et al.* (1999) for a dataset from a nearby area. However, the power-law does not appear to be a good fit. There are several distinct breaks of slope at 1, 4 and  $20 \text{ Mm}^3$ , which are also present in the ordinary magnitude-frequency distribution.

### 3.5.2 Discussion

Large slope failures occur over a range of scales, however within this range, there appear to be characteristic scales, showing that slope failures tend to have characteristic volume ranges. The multimodal magnitude-frequency plot may owe its explanation to the particular range of geological configurations in the area, or to different rock-slope stability controls (e.g., seismicity, hydrometeorological factors). The implications for landscape form and the fractal dimension cannot be determined from process observations alone. The landslides may increase the fractal dimension by imposing a “noise” on the diffusive processes; however, this noise may reduce the fractal dimension because it occurs within a finite range of scales (2.25 orders of magnitude).

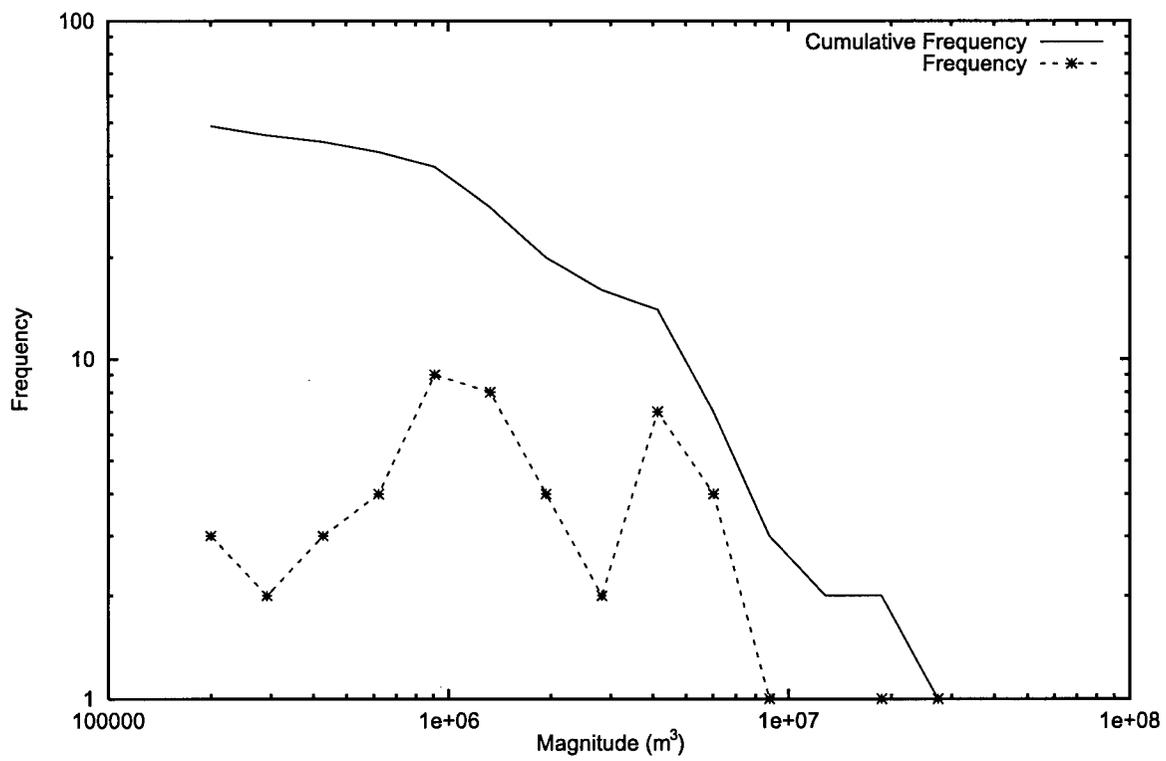


Figure 3.6: Rock Slide Magnitude Frequency (data from Savigny 1991)

## 3.6 Summary

Magnitude-frequency statistics of geomorphic processes indicate the relative importance of different sized events. In this chapter, it has been shown that on the Queen Charlotte Islands larger events are more important than the cumulative effect of small events. This has important implications for the way that these processes are incorporated into models of long-term landscape evolution: it means that small events can be treated diffusively, leaving only large mass-wasting processes to be represented explicitly.

A second use to which magnitude-frequency statistics can be put is to determine whether particular processes obey a power-law. These observations may be used to justify conjectures about the system-wide dynamics of the processes in question, and their effects on landscape morphology. Recent claims of power-law distributed landslides (Hovius *et al.*, 1997; Pelletier *et al.*, 1997) are supported by the findings of the present study from the Queen Charlotte Islands, although the power-law found here is valid over just one order of magnitude. It is premature to conclude that the slope-failures studied here are scale-invariant, although they do occur over three orders of magnitude. This does not rule out the possibility that landsliding is a power-law process that may result from the self-organization of mountain slopes to a marginally stable state (Turcotte, 1999), however it does require the restriction that this be considered a valid representation only within a limited range of topographic scales (cf. Burrough, 1993; Avnir *et al.*, 1998). The influence of landsliding on the fractal properties of hillslope topography will be the subject of the next chapter.

## Chapter 4

### Fractal properties of topography

#### 4.1 Geomorphic Processes and Fractal Landscapes

Fluvial processes are frequently thought to increase the fractal dimension of topography, whereas it is often supposed that hillslope processes reduce the fractal dimension of landscapes (see Rodríguez-Iturbe and Rinaldo, 1997, for examples of this view). In the previous chapter, hillslope processes in the study-area were shown to act over a wide range of length scales. In this chapter, digital elevation data are used to examine the relative roles of fluvial and hillslope processes in creating fractal landscapes.

The fractal dimension of a land surface is a measure of the degree to which elevation varies over different spatial scales. It is an indicator of topographic autocorrelation, and an indirect measure of topographic roughness. Fluvial features are expected to scale with upslope area because, in general, a smaller upslope area leads to a smaller river. At some point, this argument must cease to hold: i.e., for some value of upslope area the landscape no longer contains a channel. In areas of the landscape that are beyond this threshold for channel initiation, any fractal property of topography must be due to hillslope processes.

## 4.2 Fractal Properties of Landscape

Three types of fractal behaviour are widely acknowledged (see Lam and de Cola, 1993, for geographical examples):

**mono-fractal scaling:** self-similar scaling with a single fractal dimension, or self-affine scaling after an anisotropic geometric transformation (Mandelbrot, 1983);

**multifractal scaling:** characterized by an infinite number of fractal dimensions on a geometric support (first described by Frisch and Parisi, 1985, cited by Rodríguez-Iturbe and Rinaldo, 1997);

**pseudofractal scaling:** having measurable fractal characteristics but without scale-invariant cause, or scaling over only a limited range of scales (often caused by sampling limitations; Burrough, 1993).

Since the introduction of the term “fractal”, each of these concepts has been applied to the study of landscapes. River networks are examples of self-similar monofractals (Mandelbrot, 1983; Tarboton *et al.*, 1991); some topographic transects are examples of self-affine monofractals, although Veneziano and Iacobellis (1999) have suggested that scaling in the transect is not this simple. Lavallée *et al.* (1993) contend that landscapes (along with “most geographical and geophysical fields”, p. 158) are multifractal because they exhibit the mixed effects of several different processes. Responses to the claim that topography is multifractal can be divided into three groups:

- topography is a *true* multifractal because it is produced by a multifractal process, such as a multiplicative random cascade (Lavallée *et al.*, 1993; Rodríguez-Iturbe and Rinaldo, 1997).

- topography *appears* to be a multifractal because a typical landscape is produced by several processes, each imparting certain scaling characteristics to the surface. Thus non-stationarity of the fractal dimension leads to a spurious (and misleading) claim of multifractality (see Veneziano and Iacobellis, 1999).
- fractal and multifractal concepts are not useful descriptions of landscapes, because scaling occurs over only a limited range of scales and is not always related to landscape-forming processes (Burrough, 1993; Evans and McClean, 1995; Avnir *et al.*, 1998).

Veneziano and Iacobellis (1999) suggest that some parts of the landscape close to river channels do have fractal characteristics, while hillslopes do not. They show that if these two zones are mixed in fractal analysis then multifractality is observed; whereas if the two zones are analysed separately, simple mono-fractal scaling is found. They conclude that an individual watershed may be partitioned into spatial zones of homogeneous fractal dimension, based on area upslope of each point. Once these definitions have been made, elevation in the domains so created scales as a mono-fractal (i.e., with a single fractal dimension), because a single geomorphic process characteristically dominates the domain.

In the remainder of this chapter, the methods with which this result may be established are outlined, and the results of such analyses are shown for several drainage basins within the study areas described in Chapter Two.

### 4.3 Measuring Fractal Properties of Landscape

Several methods are available for estimating the fractal dimension of a topographic surface (Klinkenberg and Goodchild, 1992; Rodríguez-Iturbe and Rinaldo, 1997). These are:

- dividers method for a topographic cross-section.
- box-counting method for the boundary of a contour polygon.
- structure function analysis (variogram method).

Klinkenberg and Goodchild (1992) have performed a comparison of these methods for the estimation of fractal dimensions from digital elevation data. They conclude in favour of the variogram method because it is most stable. The variogram method will be used in the present work owing to its stability, the ease with which irregularly-spaced elevation sample points may be analysed, and the straightforward modification required to examine multifractal characteristics of the surface. The following description of this approach is based on Veneziano and Iacobellis (1999).

The fractal dimension of a sampled field of elevations may be obtained by supposing that the distribution of elevations between points at a given separation-distance is related to the separation-distance according to Equation 4.1.

$$\Delta h(L) \stackrel{d}{=} r^{-H} \Delta h(rL) \quad (4.1)$$

where  $\Delta h(L)$  is the increment in elevation between two points at distance  $L$ ,  $\stackrel{d}{=}$  denotes equality in distribution,  $r$  is some positive number, and  $H$  is a constant called the index of self-similarity. If this is the case, then the  $q$ th general moment of the distribution will behave according to Equation 4.2, for all moments that exist.

$$E[|\Delta h(L)|^q] = r^{-qH} E[|\Delta h(rL)|^q] \quad (4.2)$$

The fractal dimension,  $D = 2 - H$ , which lies between 2 and 3 for a topographic surface, may be obtained from a log-log plot of the  $q$ th moment of  $\Delta h(L)$  against separation distance  $L$ . The intercept of this plot is also useful in the present context: it is the *lacunarity* of the surface ( $\gamma_0$ ). It is independent of  $D$  and expresses the patchiness of the surface variability. A surface with a high lacunarity has many patches of different surface variability and so appears “moth-eaten”.

In addition to obtaining  $D$  and  $\gamma_0$  from the variogram plot, it is possible to examine the scaling behaviour of higher order moments of the dataset. A plot of  $H(q)$  against  $q$  will be linear if the surface is a mono-fractal, or non-linear if it is multifractal.

Data from the Queen Charlotte Islands and Greater Vancouver Regional District (GVRD) Capilano watershed were analysed using these tools, after first partitioning the surface into possible process-domains.

## 4.4 Process-domains and the Fractal Dimension

Several authors have proposed the existence of landscape areas dominated by different processes (e.g., landsliding, soil creep, solution, fluvial erosion). Ryder (1981) and Hewitt (1989) propose classifications based on altitudinal zonation of landscape processes. These observations are interesting in the present context because they are supported by claims that the fractal dimension changes with altitude (Burrough, 1993; Evans and McClean, 1995). Altitudinal zonation of landscape processes is intuitively appropriate because processes at higher altitudes have available lower concentrations of water than at lower altitudes. Reduced temperatures and different biotic processes result in a slower rate of weathering, which leads to a reduced amount of easily mo-

bilized material. Consequently, sediment transport is less frequent, or more episodic, and takes the form of surficial landslides and larger rockslides.

This spectrum of processes is summarized in the state-space diagrams of Takahashi (1981), Savage (1993), and Montgomery and Foufoula-Georgiou (1993), each of whom demonstrates that the relative concentration of water in a landscape unit is responsible for defining the dominant processes that occur within the unit (see Chapter One). While altitude and concentration of water are correlated, a more appropriate measure of (time-averaged) water concentration that may easily be approximated from digital elevation data is the area upslope of a given point.

Following Veneziano and Iacobellis (1999), landscapes may be divided into zones that are influenced by the channel network, and those that are unchanneled hillslopes. The boundary between the two is defined by examining plots of average slope against upslope area (see Fig 1.1). Mark and Aronson (1984), Tarboton *et al.* (1991) and Montgomery and Foufoula-Georgiou (1993) have each observed a break of slope at upslope areas of the order  $\sim 10^2 - 10^3 \text{ m}^2$ . It is claimed that this corresponds to the hillslope-fluvial transition; thus it is similar to the constant of channel maintenance defined by Schumm (1956).

In the present work, slope was determined from a 10 m DEM of the study area. Upslope area was found using ARC/INFO's flow accumulation algorithm, from the same elevation data. Each slope-area plot shown in this chapter demonstrates an inflection at its lower end. This was used to identify the constant of channel maintenance, allowing the hillslope and channel process domains to be separated<sup>1</sup>.

Channel network fractal dimensions were measured from the slope-area plots (Veneziano and Iacobellis, 1999), using the Octave/Matlab script `mfs_larea.m` given

---

<sup>1</sup>For the GVRD locations, slope-area plots created using this procedure were found to be similar to plots from field data (collected by White, pers comm., 2000); the two methods also produced similar estimates of the constant of channel maintenance.

in Appendix A. The fractal and multifractal characteristics of the hillslope areas were then studied using `mfs`, a C++ program also given in Appendix A. Preliminary investigation indicated that the area above the channel network was slightly multifractal when treated as a homogeneous unit. To correct for this, the hillslope region was further subdivided into an upper region (dominated by failures in bedrock) and a lower region (dominated by shallow failures in surficial material). An example of the typical zonation pattern is shown in Figure 4.1.

Results are presented below for twelve sub-basins of the Capilano River and eight drainage basins on the Queen Charlotte Islands (see Chapter Two).

## 4.5 Results

In total, twenty drainage basins were investigated. Twelve sub-basins of the Capilano River were chosen; they are underlain primarily by diorite and granodiorite with a surficial cover of glacial drift and colluvial deposits (see Chapter Two). Table 4.1 shows the area of each of these; small sub-basins were chosen in an attempt to analyse areas with consistent controlling conditions, and to preserve computational tractability in the analysis of this high-resolution data-set. On the Queen Charlotte Islands, eight watersheds were chosen in the central portion of the islands; their geologies and other data are given in Table 4.3. Of these, only in Riley, Mountain and Armentieres creeks is there a significant amount of logging (10–20% by area, according to Rood, 1990). Maps showing the location of each watershed are given in Figures 2.3 and 2.2.

### 4.5.1 GVRD Capilano Watershed

Slope-area plots for a 10 m DEM of the Capilano Watershed (supplied by GVRD Watershed Planning Department) are shown in Figures 4.2, 4.3 and 4.4. The most

common hillslope-channel threshold is 2,000 m<sup>2</sup>. This indicates that that only 2,000 m<sup>2</sup> (=0.002 km<sup>2</sup>) of contributing area is required for channel initiation in the GVRD watersheds.

Fractal dimensions of the stream networks in each of these watersheds are given in Table 4.1. The average of these is 1.33, suggesting that channel networks in this area are not as space-filling as usual (typical channel network fractal dimension is between 1.6 and 2.0; Veneziano and Iacobellis, 1999). All the DEMs studied had mono-fractal channel networks, as shown by a linear moment-slope relation (see Figures 4.2, 4.3, 4.4). The low channel fractal dimension combined with small upslope area required for channel initiation is consistent with the area's Quaternary history, the legacy of which is a mantle of highly erodible glacial deposits on which a steep, linear channel network has recently re-formed.

Watershed	Area <i>km</i> <sup>2</sup>	Slope-Channel Transition <i>m</i> <sup>2</sup>	Fractal Dimension (Network)
Daniels (1)	4.5	2000	1.39
Daniels (2)	1.1		sample too small
Daniels (3)	3.4	2000	1.52
Daniels (4)	1.0	2000	1.30
Sisters (1)	4.2	2000	1.26
Sisters (2)	6.5	2000	1.33
Sisters (3)	4.1	2000	1.23
Sisters (4)	2.1	2000	1.30
Enchantment (1)	4.3	2000	1.43
Enchantment (2)	2.0		sample too small
Enchantment (3)	1.8	2000	1.40
Enchantment (4)	2.3	2000	1.18

Table 4.1: GVRD Capilano Channel Network Fractal Dimensions

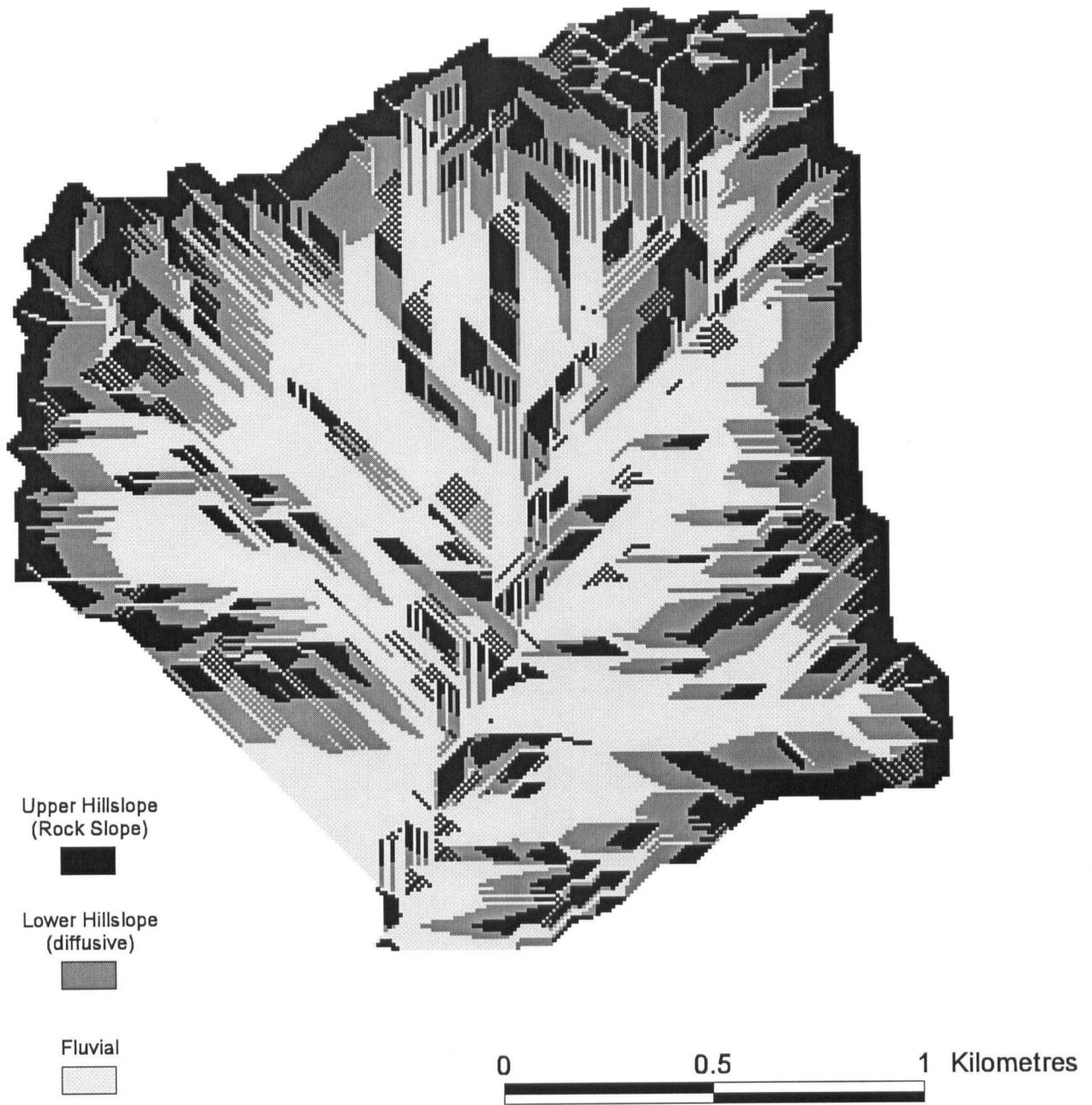


Figure 4.1: Zones for Computation of  $D$

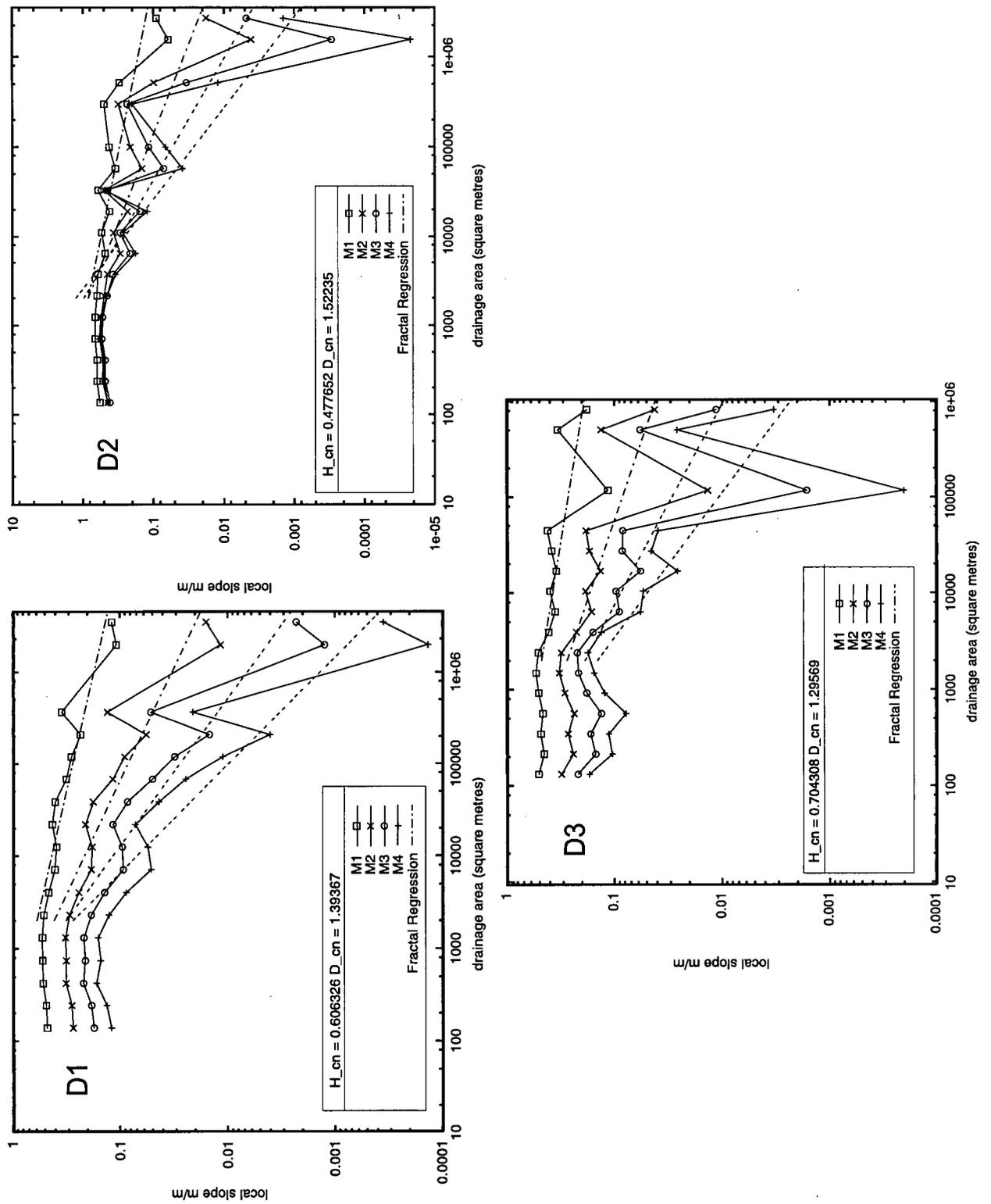


Figure 4.2: Slope-Area Plots: GVRD Daniels Creek

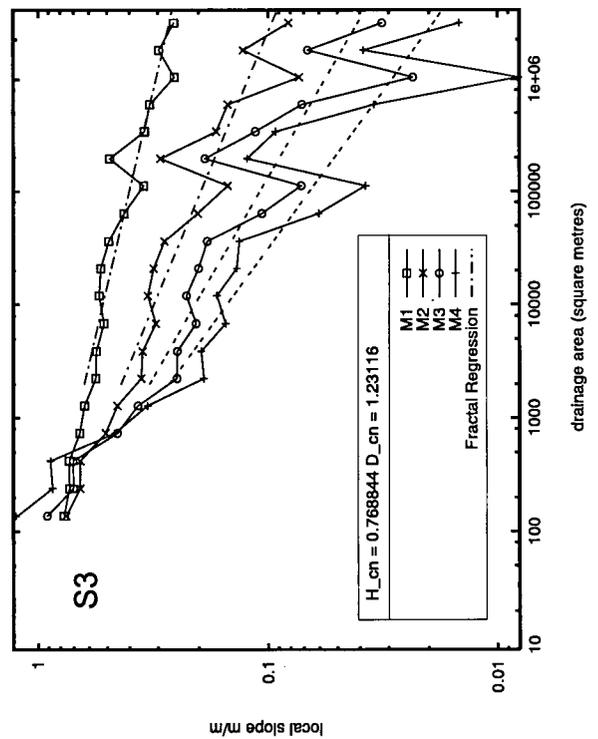
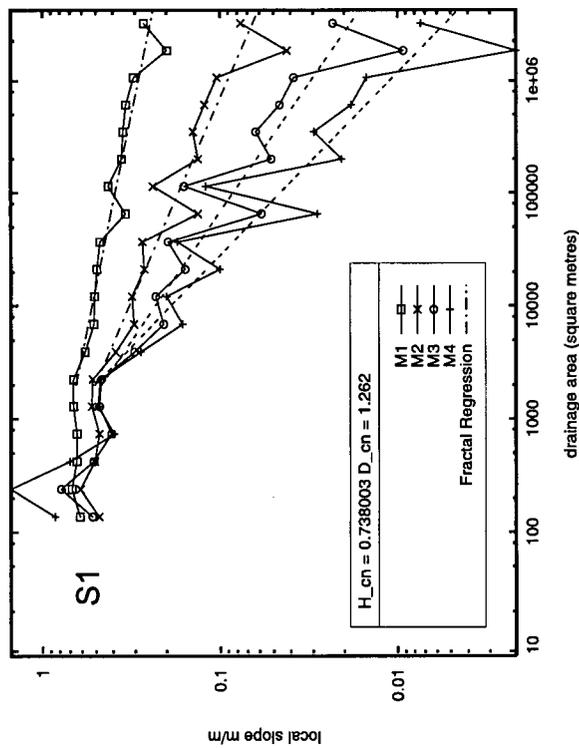
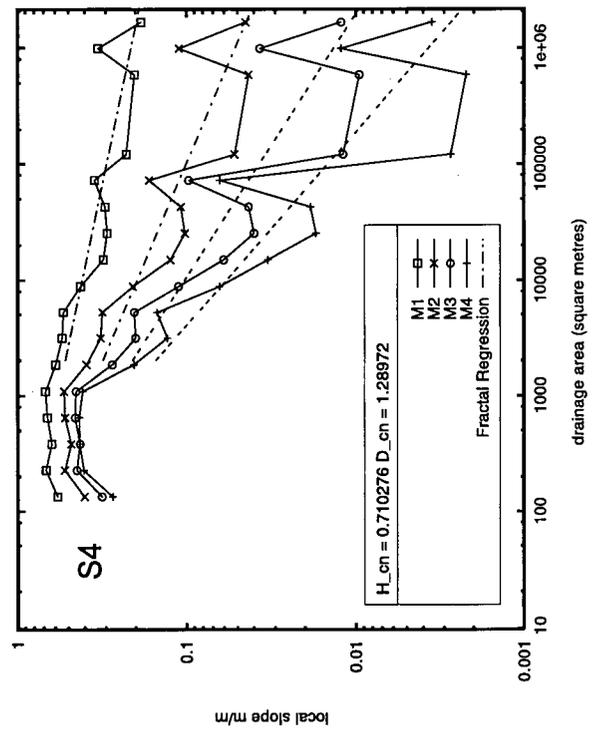
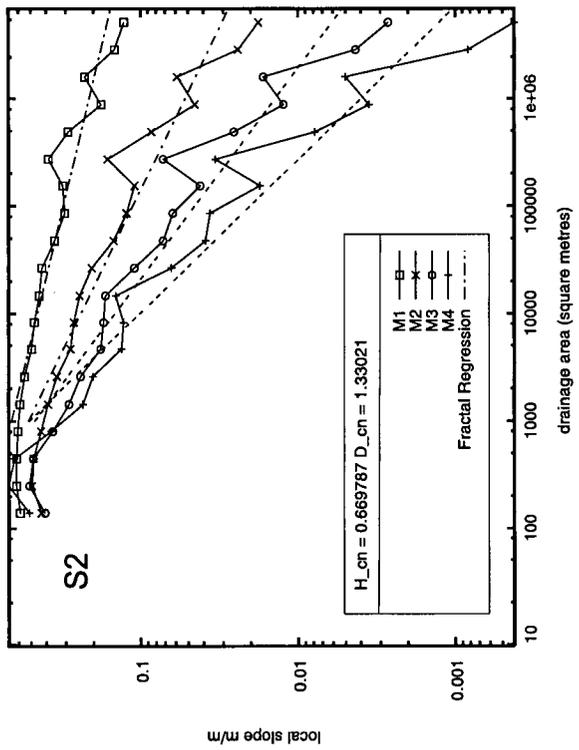


Figure 4.3: Slope-Area Plots: GVRD Sisters Creek

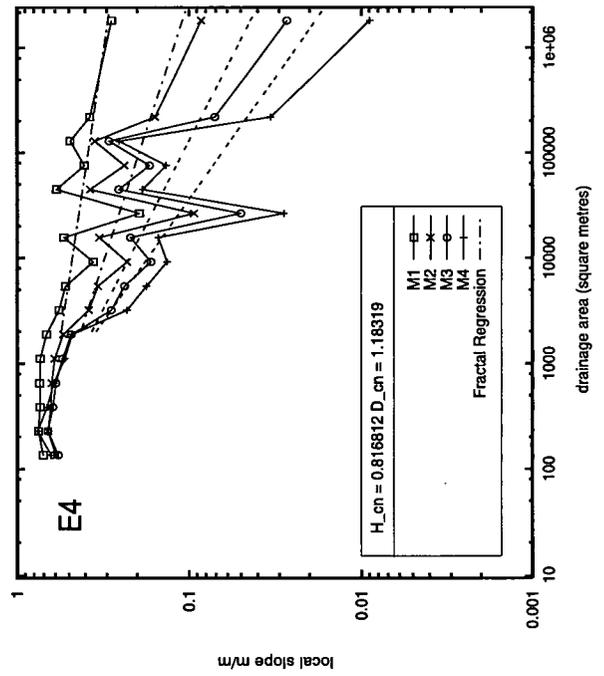
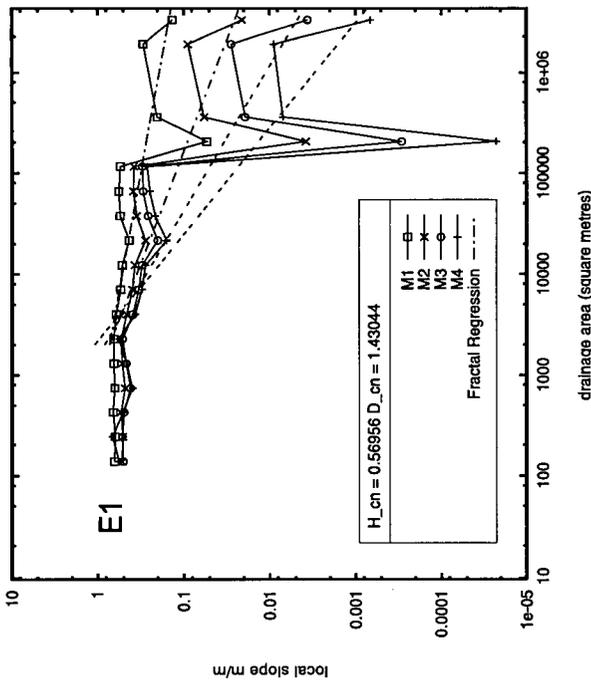
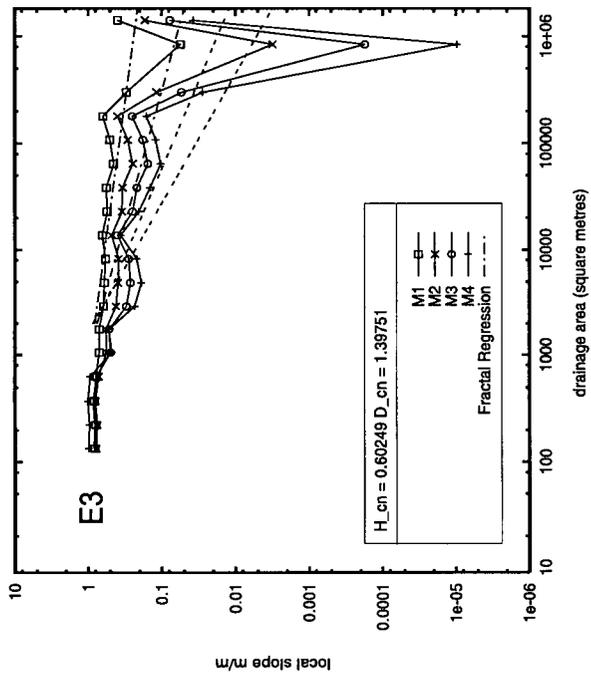


Figure 4.4: Slope-Area Plots: GVRD Enchantment Creek

Points on the channel network (i.e., with upslope area greater than 2,000 m<sup>2</sup> (20 pixels)) were removed from the elevation data in all subsequent analyses in order to avoid the problems of non-stationarity documented by Veneziano and Iacobellis (1999). To separate the upper bedrock slopes from the lower soil mantled slopes, the remaining elevation points were divided into two sets: the set  $U$ , with upslope area  $\leq 600$  m<sup>2</sup> ( $\leq 6$  pixels), and the set  $L$ , with 600 m<sup>2</sup> – 2,000 m<sup>2</sup> (6–20 pixels).

Fractal dimensions were measured using *mfs* (detailed in Appendix A) which measures fractal dimension using the variogram or structure function method. It is capable of estimating  $D$  with precision  $\pm 0.05$  for a sample of 2,000 data points. It also plots moment-slopes, which are used to determine whether or not the elevation field is multifractal (see Section 4.3). Where available, 2,000 elevation points were used; in all cases the fractal dimension reported is the average of three independent measurements of a randomly drawn subsample of the dataset in question. Variograms are plotted for separations between 10 m and 100 m (the 10 m lower limit is given by the resolution of the DEM; the upper limit is imposed to ensure that sample points remain close to the same hillslope, following the advice of Veneziano and Iacobellis 1999). The resulting variograms are shown in Figures 4.5 through 4.10 and the results are summarized in Table 4.2.

The results show that in all cases  $D$  is significantly greater than 2.0, indicating that there is scaling (over a limited range of one order of magnitude) for the hillslope regions. In all cases,  $D$  for upper hillslopes is greater than or equal to that for lower hillslopes (although it remains unclear whether the difference is within the resolution of the algorithm for estimating  $D$ ). In all cases but one, the scaling was mono-fractal (as shown by the linear increase of higher order moment-slopes in Figures 4.5 through 4.10). The one case where a slight multifractality was noticed is for Sisters Creek (3). This creek has a particularly low channel network fractal dimension (cf. drainage density),

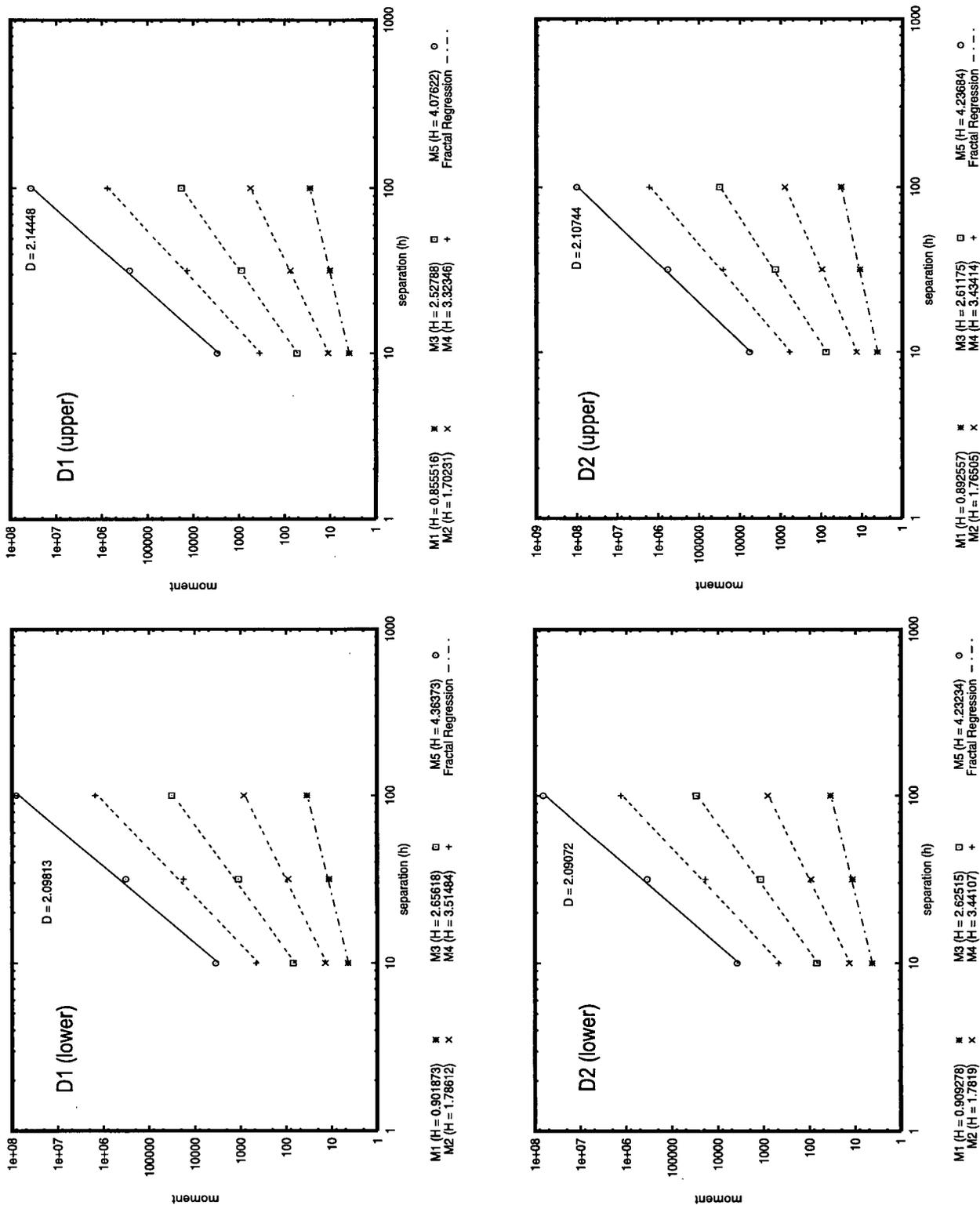


Figure 4.5: Hillslope Variograms: GVRD Daniels Creek (D1 – D2)

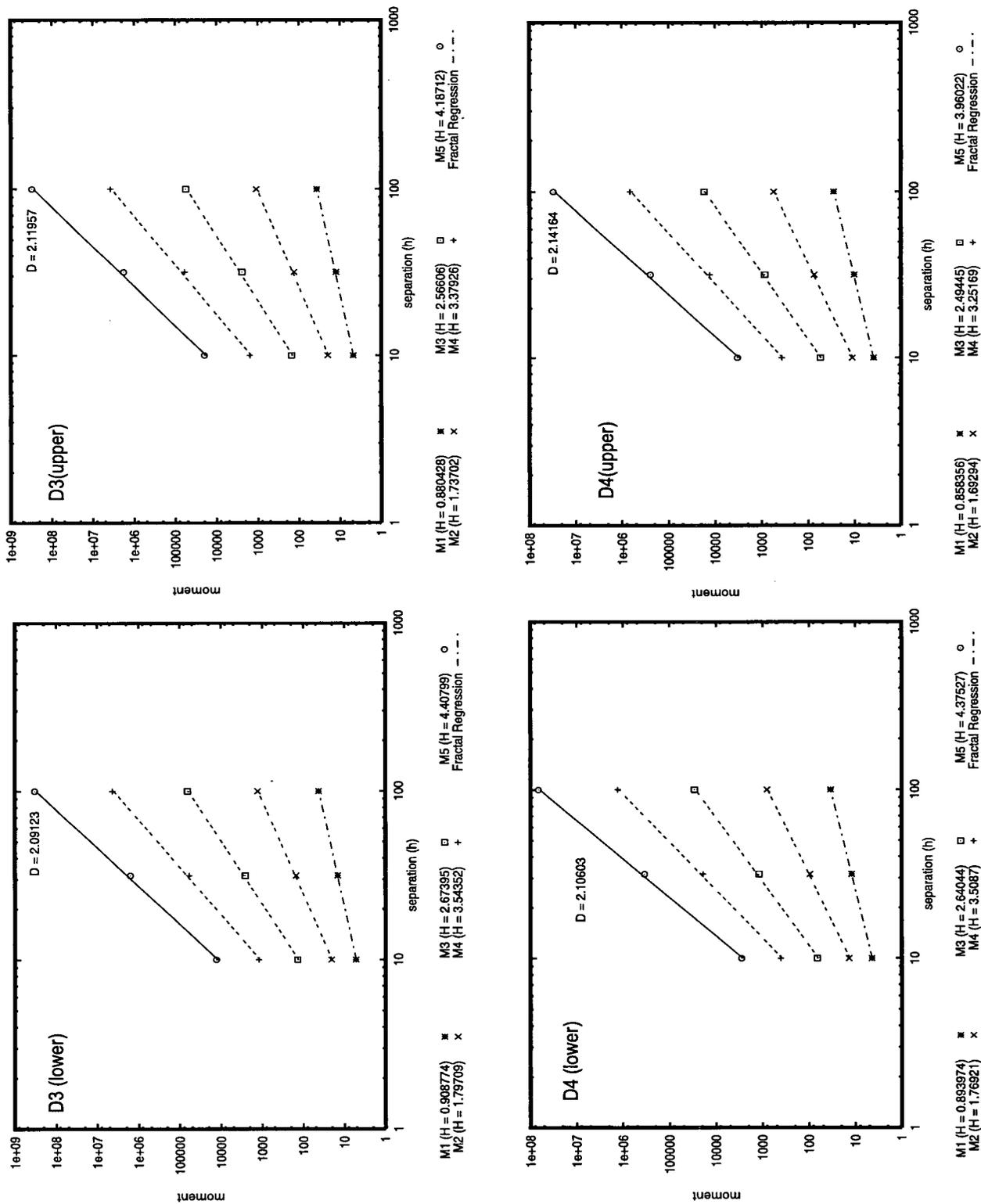


Figure 4.6: Hillslope Variograms: GVRD Daniels Creek (D3 – D4)

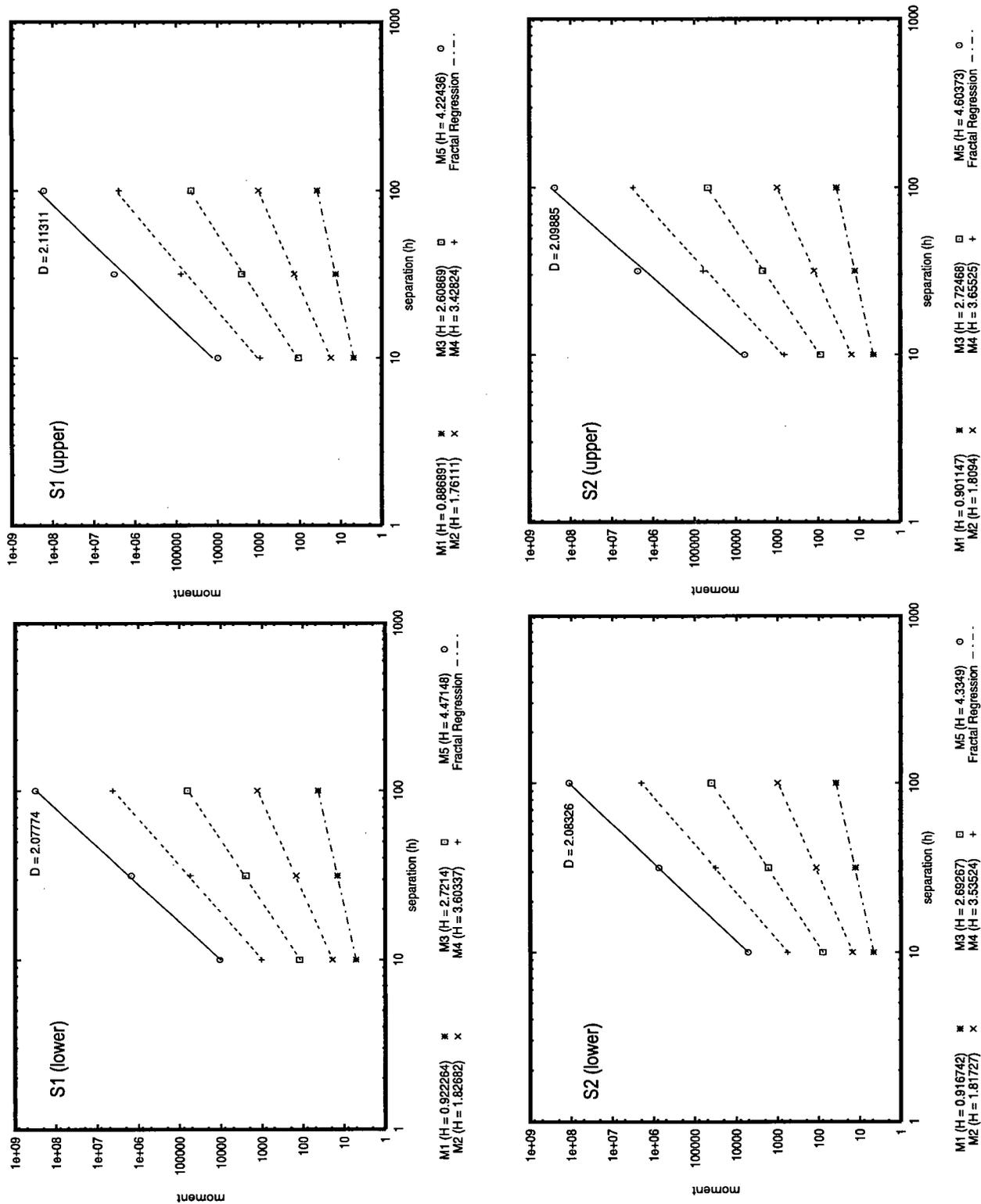


Figure 4.7: Hillslope Variograms: GVRD Sisters Creek (S1 – S2)

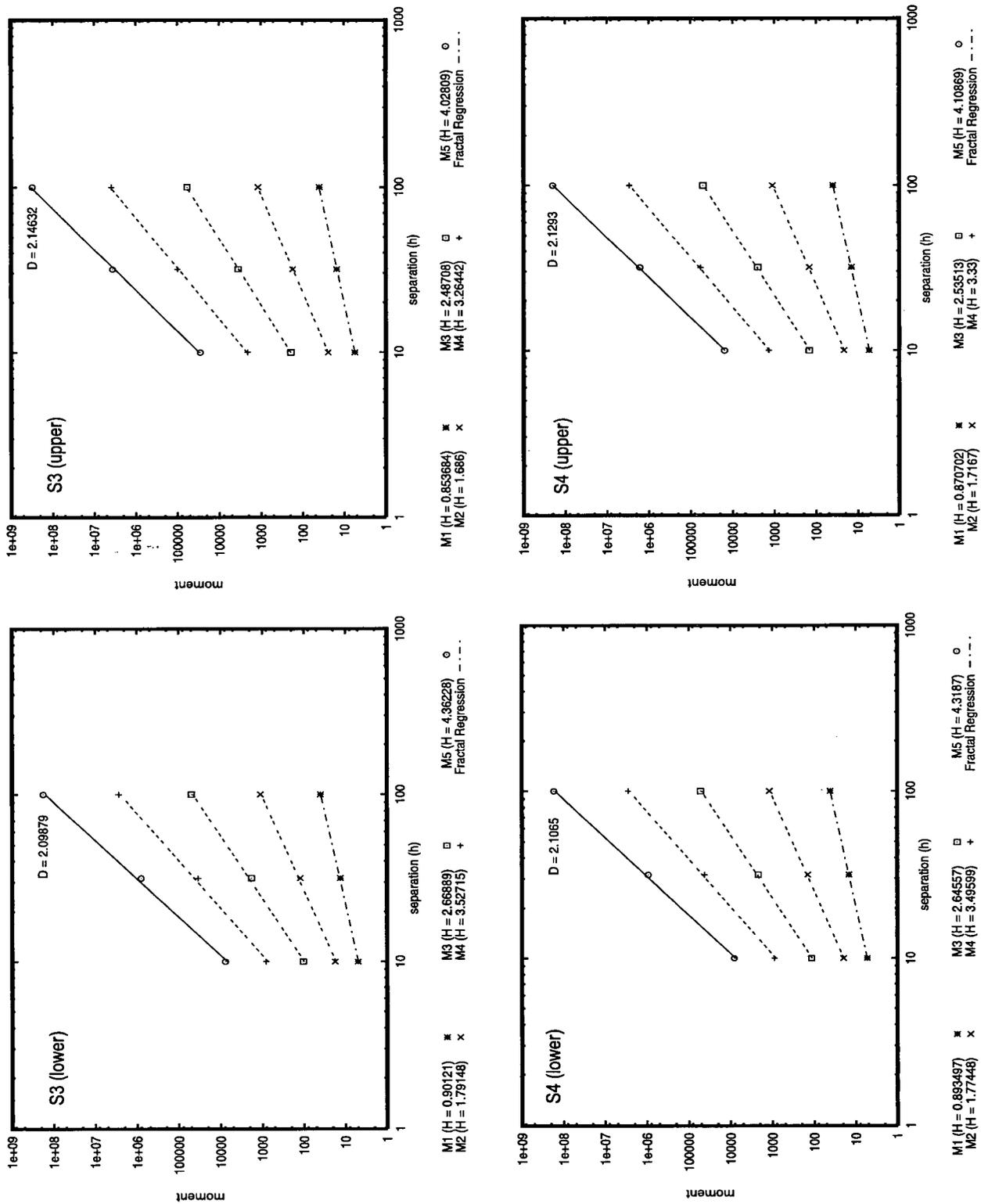


Figure 4.8: Hillslope Variograms: GVRD Sisters Creek (S3 – S4)

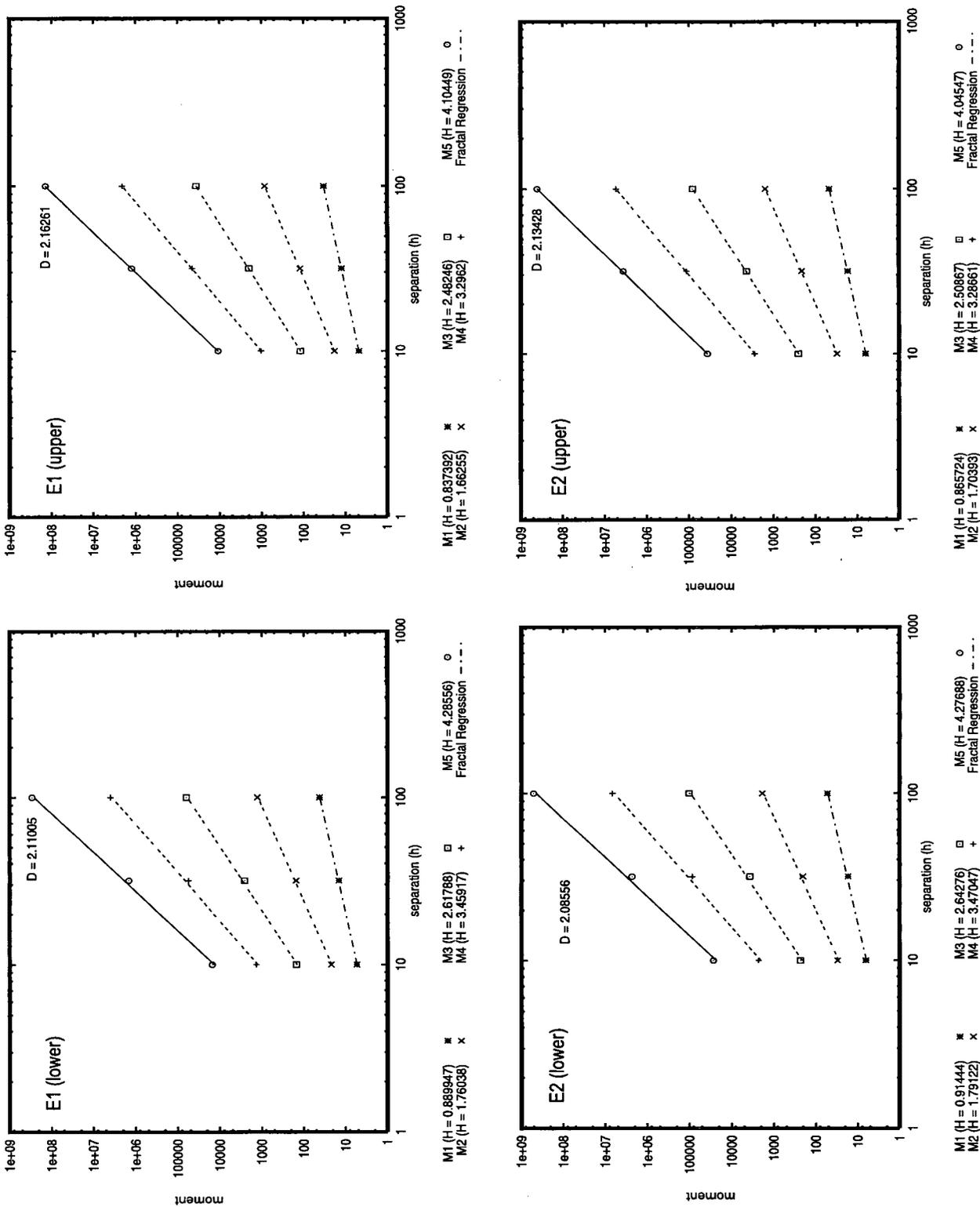


Figure 4.9: Hillslope Variograms: GVRD Enchantment Creek (E1 – E2)

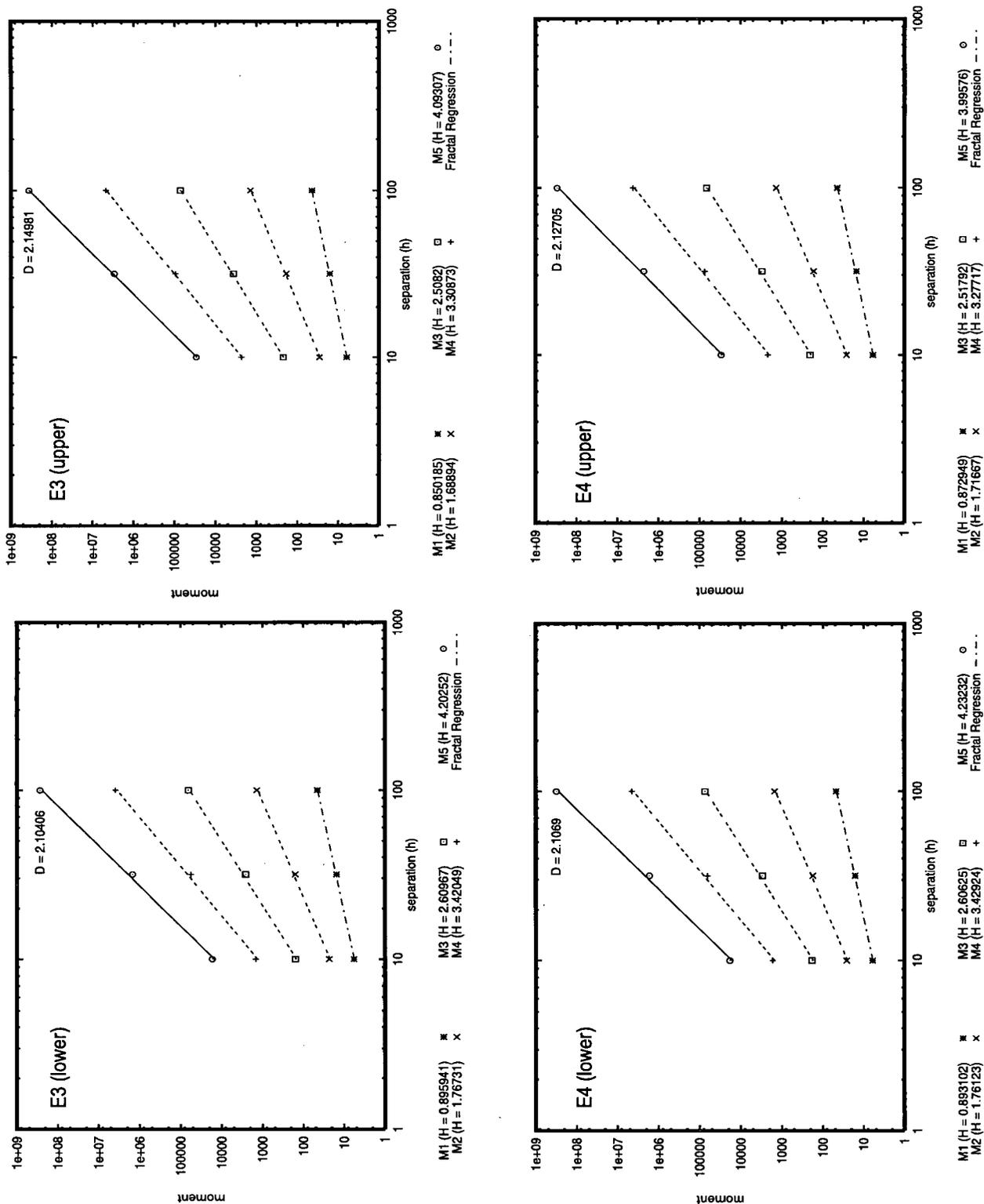


Figure 4.10: Hillslope Variograms: GVRD Enchantment Creek (E2 – E3)

Watershed	Lower Slopes		Upper Slopes	
	$D$	$\gamma_0$	$D$	$\gamma_0$
Daniels (1)	2.10	-0.26	2.15	-0.27
Daniels (2)	2.10	-0.27	2.11	-0.28
Daniels (3)	2.10	-0.17	2.12	-0.20
Daniels (4)	2.10	-0.27	2.14	-0.27
Sisters (1)	2.10	-0.19	2.13	-0.15
Sisters (2)	2.07	-0.27	2.11	-0.22
Sisters (3)	2.09	-0.22	2.14	-0.10
Sisters (4)	2.10	-0.17	2.12	-0.16
Enchantment (1)	2.12	-0.16	2.17	-0.14
Enchantment (2)	2.08	-0.12	2.13	-0.07
Enchantment (3)	2.11	-0.14	2.15	-0.06
Enchantment (4)	2.10	-0.14	2.13	-0.11
Average	2.10	-0.20	2.13	-0.17

Table 4.2: GVRD Capilano Hillslope Fractal Statistics

indicating that hillslope and channel processes may be less distinct than in areas with a more extensively developed channel network. The appearance of multifractality may therefore have resulted from inhomogeneity of the processes occurring within this sub-basin.

The Capilano sub-basins have non-planar hillslopes. This is contrary to the common assumption that hillslopes are dominated by diffusive, homogenizing processes that destroy fractal properties introduced by fluvial processes. Furthermore, in some cases there is a distinction between the “rougher” upland hillslopes that may be dominated by rarer events and the lower hillslopes, which are dominated by more frequent shallow failures leading to a lower fractal dimension.

The results presented here suggest that there is some characteristic of hillslope mass-wasting that modifies  $D$  in areas not influenced by stream channels. This may be a sediment transport process that operates over a wide range of length scales (such as the hillslope processes described in Chapter Three); alternatively, it may be

a scale-bound process that leads to a pseudo-fractal effect captured in this analysis (see Burrough, 1993). Since this question is relevant to both the GVRD Capilano watersheds and the Queen Charlotte Islands, it will be discussed after the presentation of results from the latter.

#### 4.5.2 Queen Charlotte Islands

Slope-area plots for eight watersheds on the Queen Charlotte Islands are shown in Figures 4.11 and 4.12. (These were created from a 30 m resolution grid interpolated from BC TRIM data using the ARC/INFO TOPOGRID routine.) The most common hillslope-channel threshold is  $6,000 \text{ m}^2$  ( $= 0.006 \text{ km}^2$ ). This value is three times higher than the hillslope-fluvial transition in the Capilano sub-basins, suggesting that there is a difference in the flow-properties and erodibility of the substrate<sup>2</sup>.

The fractal dimensions of the channel networks in the Queen Charlotte Islands watersheds are given in Table 4.3. These are considerably higher than those for the GVRD, indicating that the channel networks are better developed on the Queen Charlotte Islands, since they fill the plane more completely. They are all mono-fractals, as shown by the linear relation between moment-slopes for higher order moments. The combination of the greater upslope area required for channelized flow and higher fractal dimension of streams on the Queen Charlotte Islands indicates that these landscapes are more permeable and more highly erodible than those of the GVRD Capilano watershed, which is consistent with the geology and glacial history of the two areas (the Queen Charlotte Islands have had longer ice-free periods during which channel network development has occurred; see Chapter Two).

---

<sup>2</sup>The hillslope-channel threshold increased three-fold when the DEM resolution decreased by the same proportion. In the present case, this is considered to be genuine, rather than an artifact of the flow accumulation routine, since that algorithm does not use information on cell sizes.

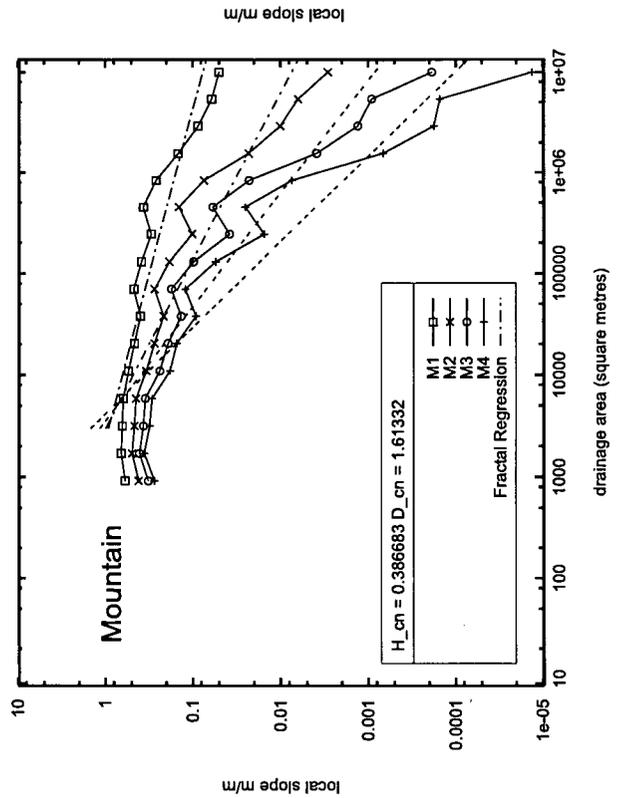
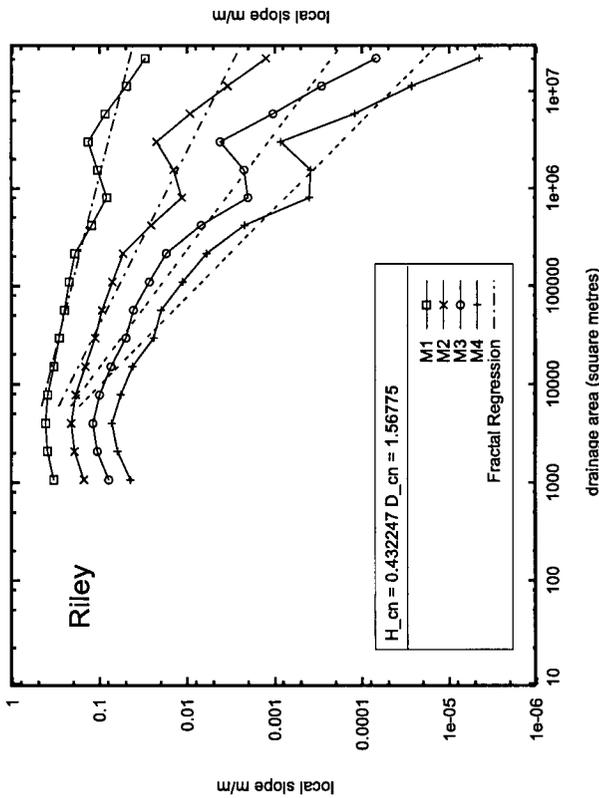
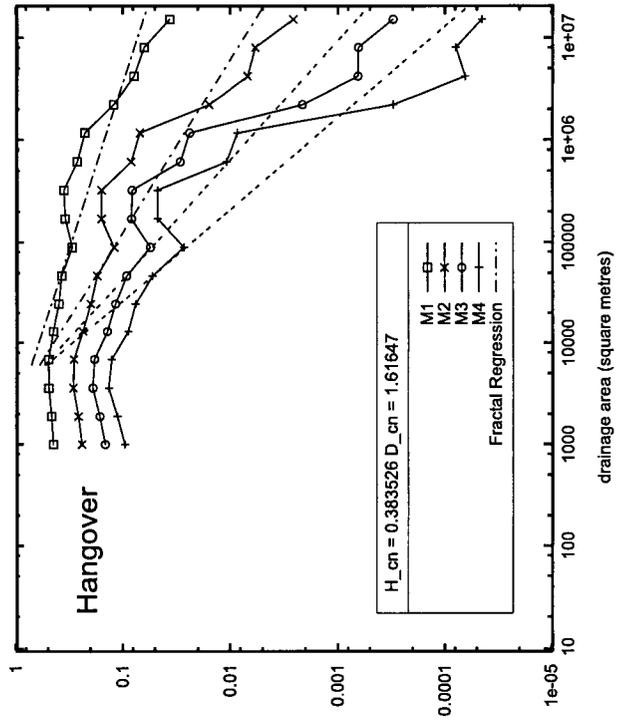
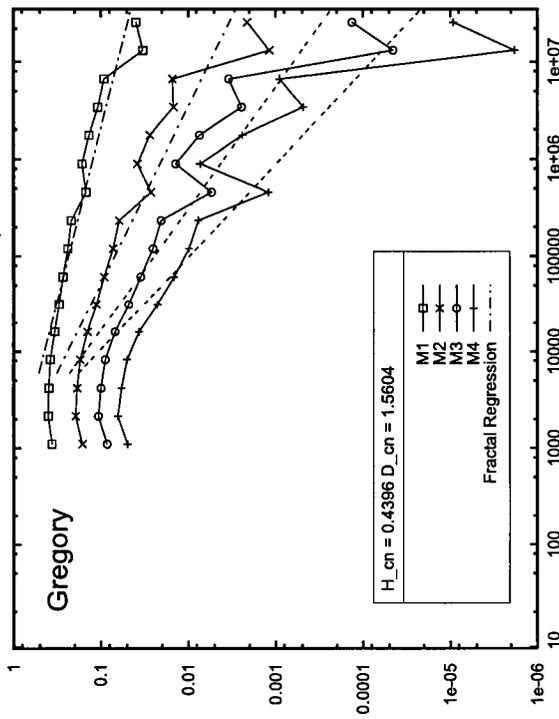


Figure 4.11: Slope-Area Plots: Queen Charlotte Islands (Rennel Sound Area)

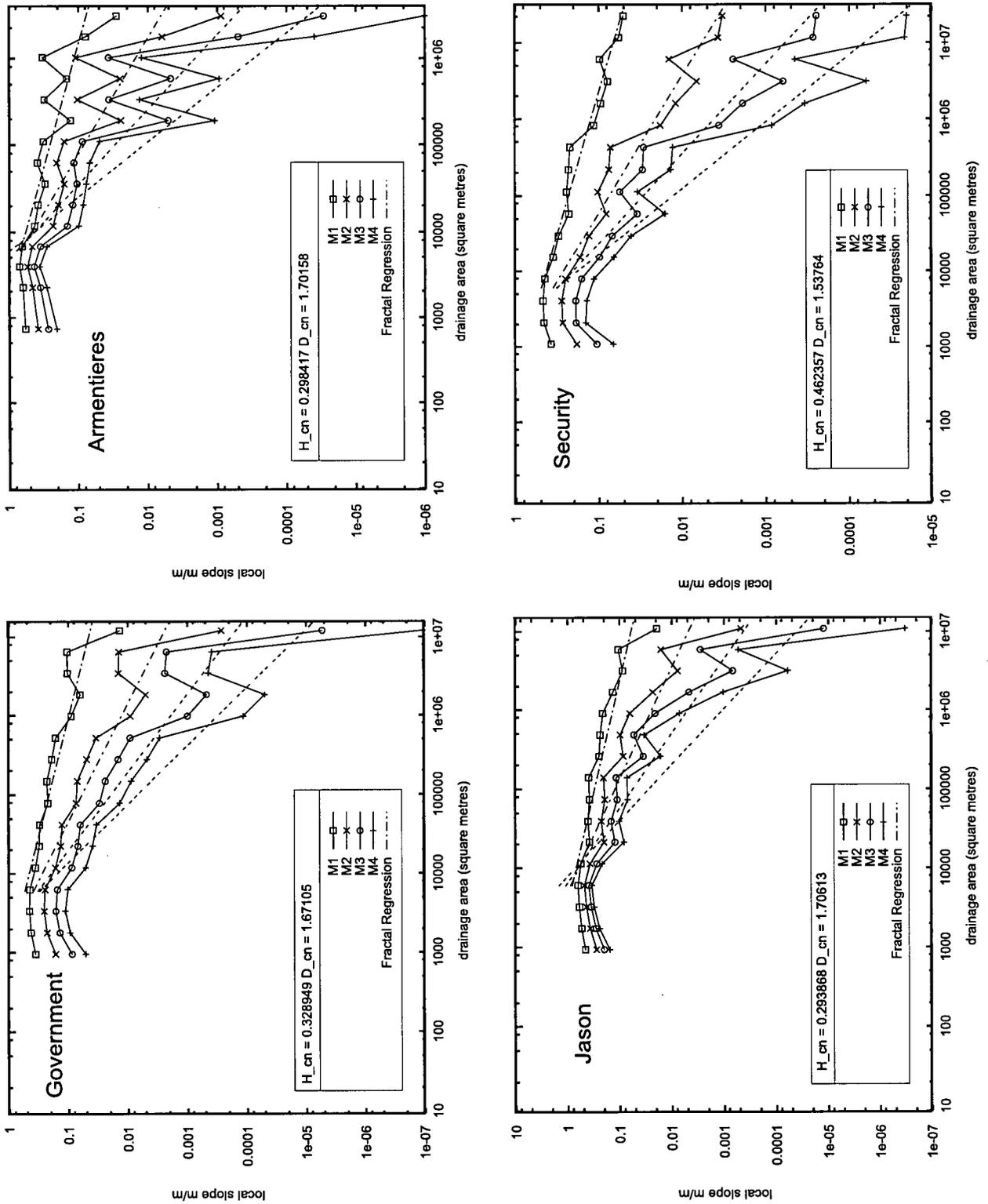


Figure 4.12: Slope-Area Plots: Queen Charlotte Islands (Skidegate Channel Area)

Watershed	Area <i>km</i> <sup>2</sup>	Physiography	Major Geological Formation	Slope-Channel Transition <i>m</i> <sup>2</sup>	Fractal Dimension (Network)
Riley	28.7	SP	Yakoun	6000	1.57
Gregory	36.7	SP	Masset	6000	1.56
Hangover	21.2	SP	Masset	6000	1.62
Mountain	12.8	QCR	PTP	3000	1.61
Government	16.1	QCR	Karmutsen	6000	1.67
Armentiere	4.0	QCR	Karmutsen	6000	1.70
Jason	11.9	QCR	Karmutsen	6000	1.71
Security	33.9	QCR	Karmutsen	6000	1.54

Table 4.3: Drainage Basin Characteristics and Channel Network Fractal Dimensions: Queen Charlotte Islands (compiled from Rood, 1984)

Key: SP=Skidegate Plateau; QCR = Queen Charlotte Ranges; PTP = Post-tectonic Plutons (see Chapter Two)

The procedure for zoning elevations by upslope area was carried out: fluvial points with upslope area  $\geq 6,000 \text{ m}^2$  ( $\geq 6$  pixels) were removed; those pixels remaining were divided into two zones: upper hillslopes with area  $\leq 1,800 \text{ m}^2$  ( $\leq 2$  pixels) and a lower hillslope zone with  $1,800 \text{ m}^2 < \text{upslope area} \leq 6,000 \text{ m}^2$ . These zones formed clip regions for the original BC TRIM spot heights, whose fractal dimensions were then analysed using *mfs*, with a separation range 60 m – 600 m (again, the lower limit set by the data resolution and the upper limit imposed to ensure that hillslope boundaries were not crossed, following Veneziano and Iacobellis 1999).

The resulting variograms and moment-slope plots are shown in Figures 4.13 through 4.16 and summarized in Table 4.4. In all cases the fractal dimension was significantly greater than two, which is the base value that would be expected for planar slopes resulting from diffusive processes. In general, hillslope fractal dimensions on the Queen Charlotte Islands are higher than in the Capilano watershed, which is consistent with the wider range of mass-movement events and more rapid rates of sediment flux presented in Chapter Three.

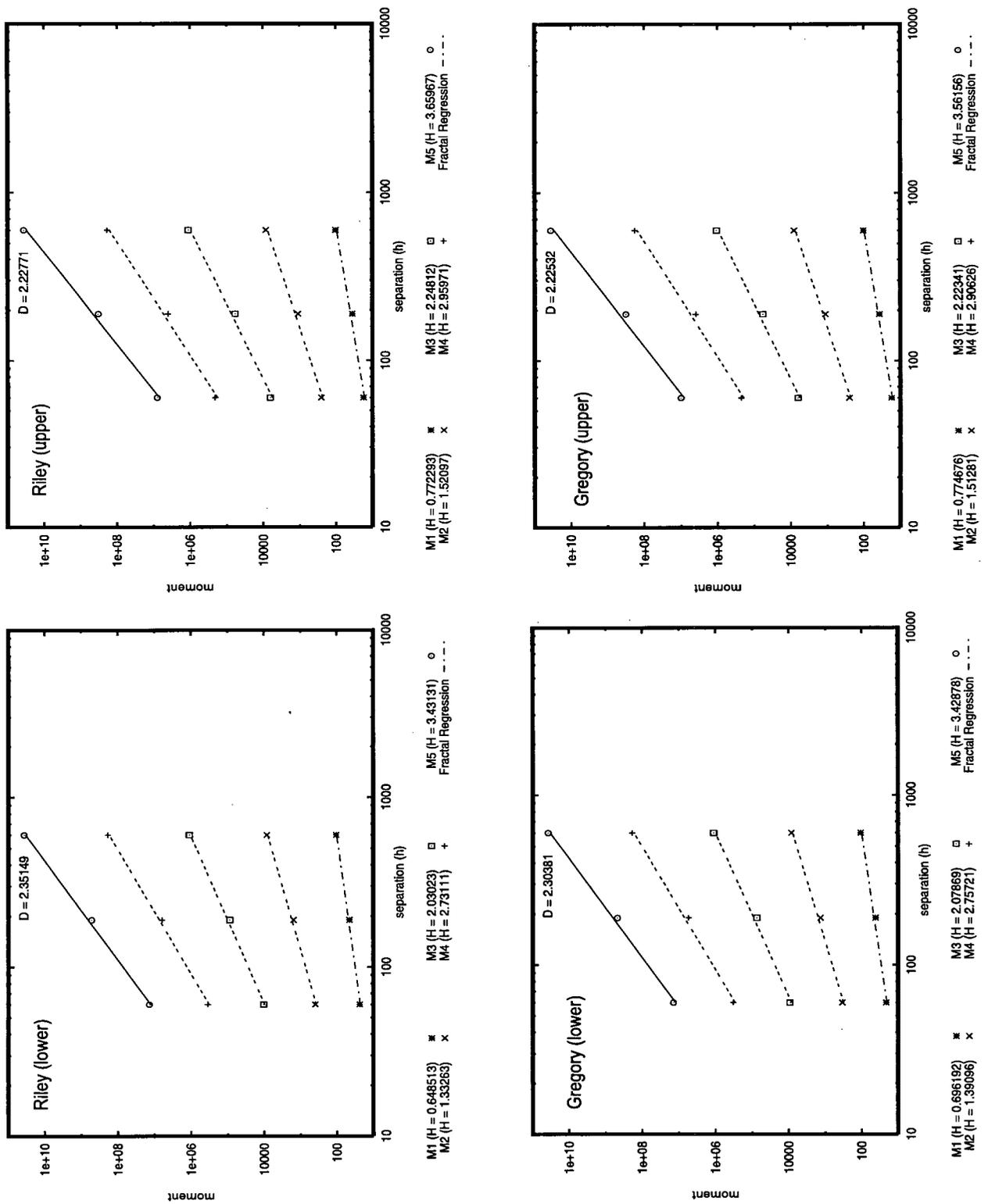


Figure 4.13: Hillslope Variograms: Queen Charlotte Islands (Rennel Sound: Riley and Gregory Creeks)

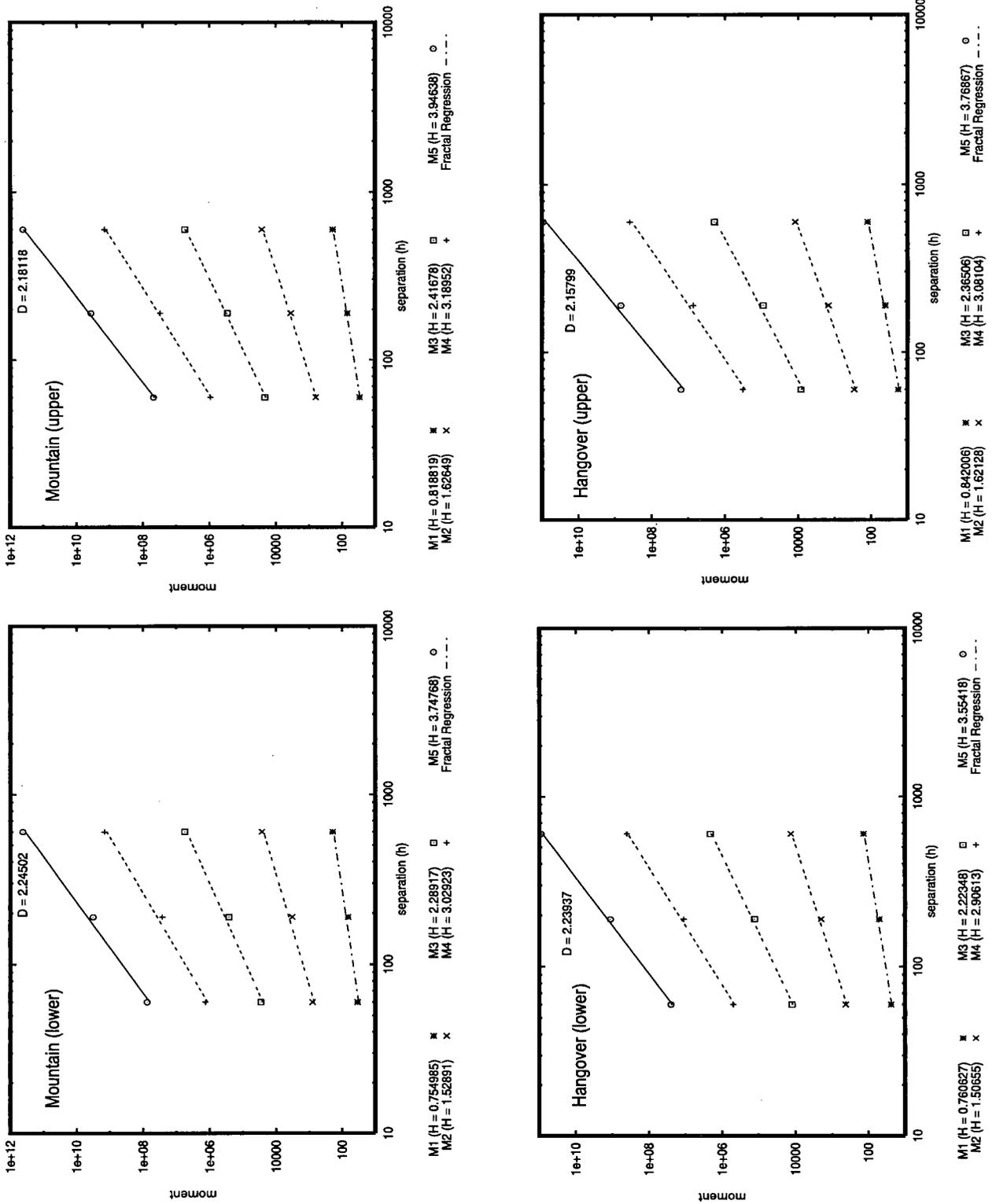


Figure 4.14: Hillslope Variograms: Queen Charlotte Islands (Rennel Sound: Mountain and Hangover Creeks)

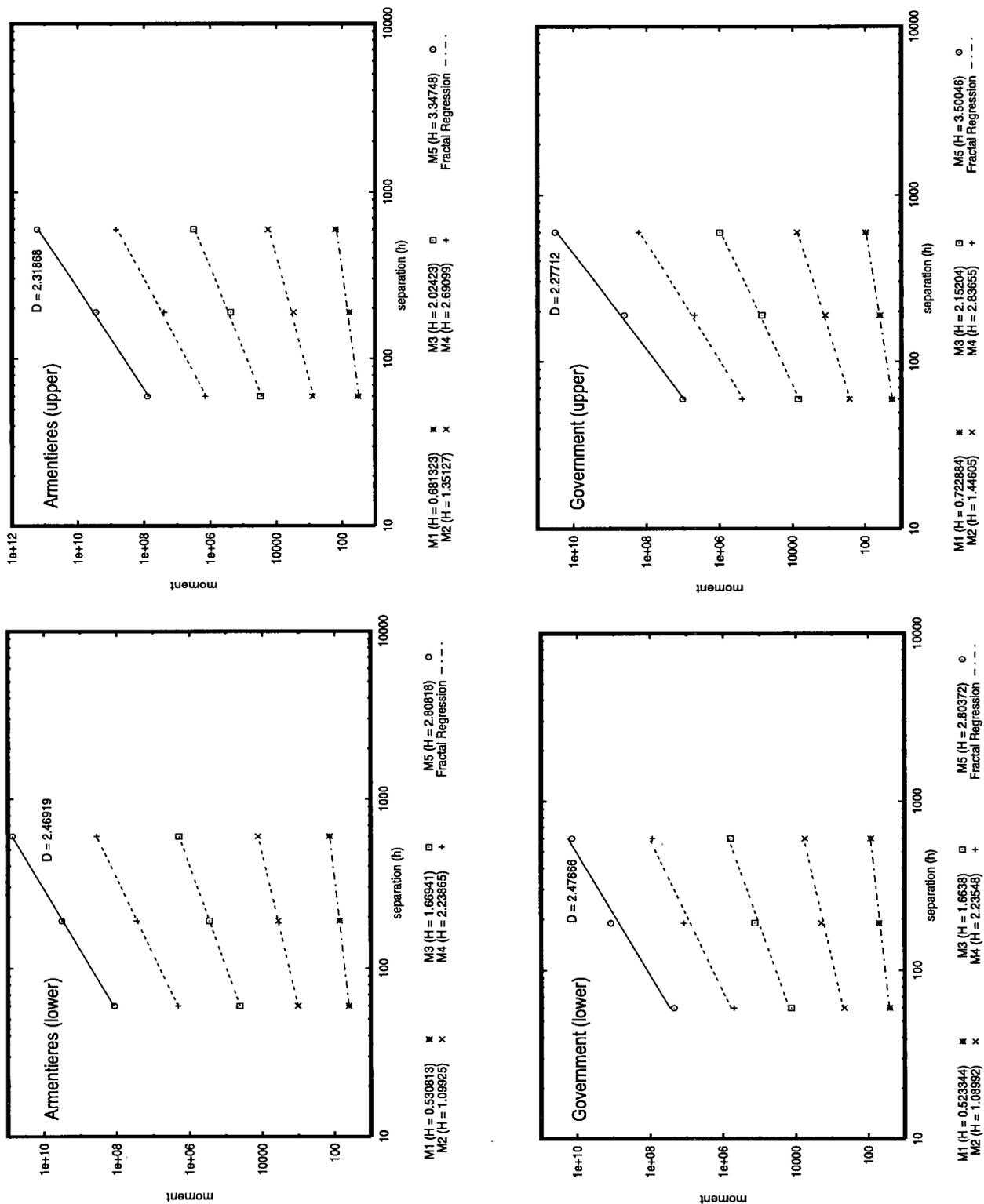


Figure 4.15: Hillslope Variograms: Queen Charlotte Islands (Skidegate Channel: Armentieres and Government Creeks)

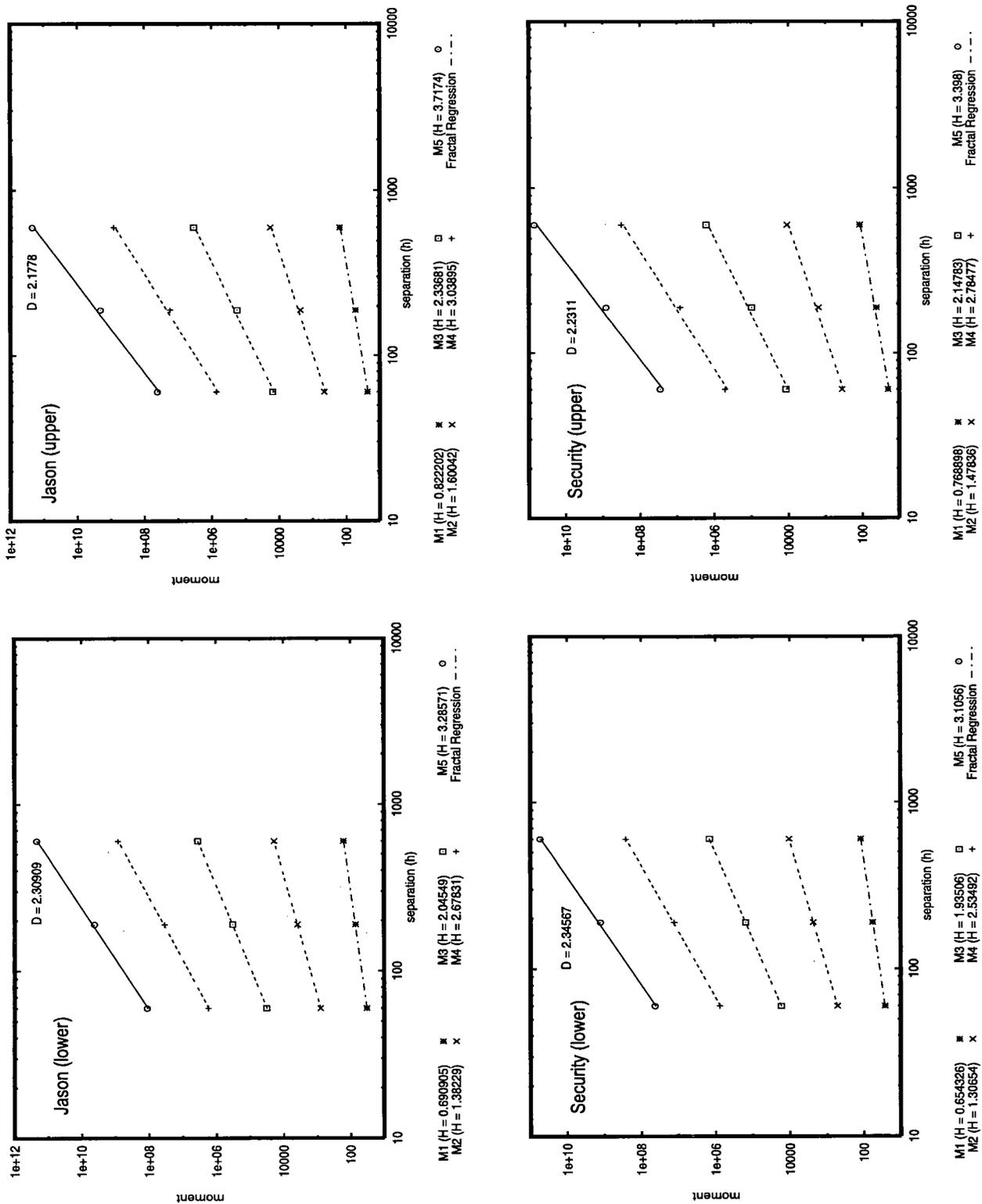


Figure 4.16: Hillslope Variograms: Queen Charlotte Islands (Skidegate Channel: Jason and Security Creeks)

Watershed	Lower Slopes			Upper Slopes		
	N	$D$	$\gamma_0$	N	$D$	$\gamma_0$
Riley	1314	2.35	-0.20	2000	2.22	-0.17
Gregory	1818	2.30	-0.07	2000	2.23	-0.15
Mountain	805	2.25	-0.16	1368	2.18	0.00
Hangover	1027	2.24	0.01	1711	2.15	-0.27
Government	866	2.48	0.49	1449	2.28	-0.04
Armentiere	250	2.47	0.66	309	2.32	0.28
Jason	723	2.31	0.28	1288	2.17	-0.12
Security	1604	2.35	0.26	2000	2.23	-0.10
Average		2.34	0.16		2.22	-0.07

Table 4.4: Queen Charlotte Islands Hillslope Fractal Statistics

There is an important difference between the characteristics of the Queen Charlotte Islands slopes and those in the Capilano watershed: without exception, the upper hillslopes on the Queen Charlotte Islands have a lower fractal dimension than the lower hillslopes. For some watersheds the difference is less than the precision of the measurement, but for many it is clear. There may be several explanations for this result, which is counter-intuitive and opposite to the GVRD result:

1. there is a lower incidence of large structure-generating landslides on the Queen Charlotte Islands than in the GVRD mountains. This is supported by Alley and Thompson (1978) and Ryder (pers. comm.), who have found only localized areas of deep-seated bedrock landsliding on the Queen Charlottes.
2. weathering rates are higher on the Queen Charlotte Islands, so slide evidence is obliterated more rapidly.
3. there is more hillslope activity on the lower slopes of the Queen Charlotte Islands. However, the hillslope processes expected to act on these slopes are predominantly shallow landslides, which have been assumed to smooth the surface over

long timescales. Their cumulative effect is considered to be diffusive, so they ought to reduce the fractal dimension rather than increase it. The increased fractal dimension observed in this process-domain may be a result of the “noise” generated by individual landslides. The smoothing effect of diffusion is revealed only after a long time; meanwhile, individual landslides (which are power-law distributed over a finite range) increase the fractal dimension.

4. human activity (logging) has increased the tendency for midslope regions to fail, leading to a higher  $D$  in those areas.

Of these explanations, (1), (2) and (3) seem the most realistic, since (4) implies that its effect will be seen only in areas with high areal proportion of logging, which is not the case. Point (3) is significant because it implies that, while it may correctly reproduce process rates, diffusion may be less successful at reproducing the resulting landscape form.

A further explanation which may be important in the present context relates to the different glacial history of the Queen Charlotte Islands (see Chapter Two). This region was not fully glaciated, leaving higher elevations ice-free. Conversely, the GVRD area was fully glaciated. The absence of ice from upper parts of the Queen Charlotte Islands landscape may therefore be responsible for the lower fractal dimension seen there, compared with the glaciated lowlands.

## 4.6 Discussion

The two areas studied in this chapter share several fractal characteristics: the most elementary is that hillslopes in the study area have a fractal dimension consistently higher than 2.0, which is the base dimension that would be expected from planar

slopes. This is contrary to the widespread view that hillslope processes destroy the fractal properties of topography. The present work has found that hillslope processes have distinct fractal characteristics when studied separately from channel networks. This evidence supports the view of Veneziano and Iacobellis (1999) that confounding hillslope and channel behaviour leads to multifractal scaling similar to that observed by Lavallée *et al.* (1993) and Rodríguez-Iturbe and Rinaldo (1997).

It is at present unclear whether the hillslope part of the landscape is truly fractal. It does not appear to have multifractal characteristics in this study, as shown by the linear relation between moment order and moment slope for higher general moments of the variogram. However, the possibility remains that the hillslope region is a pseudo-fractal, owing to the small range over which the observations have been shown to scale (one order of magnitude). If this is the case then the fractal appearance of the landscape should be interpreted as a finite scaling effect.

A distinction can be made between the findings of Veneziano and Iacobellis (1999) and those of the present study. Veneziano and Iacobellis found mono-fractal channel networks and hillslope regions with  $D$  close to 2.0. In the areas studied here, a mono-fractal channel network was found, but  $D$  for the upper and lower hillslope regions is significantly different from 2.0. This implies that in this environment (a recently glaciated, collisional orogen), there is a separate process responsible for increasing the hillslope fractal dimension.

It is suggested here that power-law-distributed landslides may be responsible for the observed fractal properties of hillslopes. Over sufficiently long timescales the effect of shallow landslides may be generalized, yielding a slope-dependent diffusive transport relation (e.g., Culling, 1960; Martin and Church, 1997). However, for instantaneous digital elevation samples, the effects of the small-scale process that make up the diffusive behaviour may introduce detectable noise (as in the case of the Queen

Charlotte Islands). The effect of large events such as deep-seated rockslides is more permanent: they cannot be treated diffusively unless the timescales of the study are very long ( $\sim 10^6$  a –  $10^9$  a).

In the final chapter of this thesis, these conjectures will be tested using a numerical model of landform evolution that incorporates hillslope diffusion, fluvial erosion and stochastic deep-seated bedrock landsliding.

## Chapter 5

# Synthesis: Process-Based Modelling with Episodic Events

### 5.1 Introduction

The formulation of integrated, multiple-process models of the long-term evolution of landscapes has been a recurrent goal in geomorphology (Davis, 1899; Penck, 1924; Culling, 1960; Merritts and Ellis, 1994; Martin, 1998; Beaumont *et al.*, 2000). A recent trend has been to develop models that incorporate generalized physical representations of measurable small-scale processes, seeking to demonstrate their cumulative effects over much longer timescales. This is an attractive approach because it allows models to be calibrated with respect to time at the outset, a feature that was deficient in earlier models (especially those of Davis).

The use of a landscape evolution model in the present work is to simulate the projected effects of measured rates of episodic processes over timescales longer than the available record. In addition, it is possible to control the processes incorporated in the model in order to decide more conclusively which are responsible for the resulting structure of the landscape. In this chapter the use of such a model is described. Landscape evolution equations are formulated and solved numerically using a computer, in

order to determine the effects of various landscape processes (episodic and otherwise) on the fractal dimension of topography. This is a strictly uniformitarian approach, since it assumes that process rates calibrated from contemporary field observations will persist over the timescales of interest.

## 5.2 Constructing Testable Landscape Models

A numerical model of landform evolution that includes several processes is a type of complex hypothesis (Oreskes *et al.*, 1994); the scientific use of this type of model is therefore worthy of discussion. The following example serves to illustrate the problems involved in testing a multiple process landform evolution model:

Suppose that a geomorphologist constructs a model (complex hypothesis) of the form: “there are processes  $P$ ,  $Q$ ,  $R$ , and  $S$ , which are linked in the way  $T$ , and which produce the outcome  $X$ , where  $X$  is an observable property of the resulting landscape”. (For clarity, one may suppose that  $P$  is soil creep,  $Q$  is shallow landsliding,  $R$  is fluvial erosion,  $S$  is rock sliding and  $T$  is a means of linking the processes such as a continuity equation or sediment budget.  $X$  is a measurable geomorphometric property such as the average slope, hypsometry or fractal dimension that arises from the modelled processes.)

Verification of a model such as this conventionally involves a demonstration of its truth (i.e., that it is a good representation of reality), most obviously through a correct prediction of the property  $X$ . It is widely acknowledged that a model or hypothesis is not *conclusively* verified by a correct prediction; its truth is merely made more likely. Falsification of a theory, model or hypothesis is generally taken to have a more conclusive effect on the device’s perceived veracity (Haines-Young and Petch, 1986, give a clear illustration of these claims in the context of physical geography).

However, as Oreskes *et al.* (1994) explain, falsification is also problematic because even simple single-hypothesis models are never closed systems; they rely on theoretical terms that are laden with inferences and assumptions. As a consequence, any failure of the model to agree with empirical observations may be explained and eliminated by modification of an auxiliary hypothesis or initial condition. This is an example of the problem of the underdetermination of theories by evidence (i.e., the Duhem-Quine thesis, for discussion of which see Duhem 1905, Quine 1953).

In short, it is not possible to devise a critical experiment. At most, one can enumerate all the recognized processes that might be responsible for creating fractal topography and construct sufficiently critical tests *between* them, ultimately to arrive at a tentative claim that one process combination is more (or less) successful at producing the relevant observed feature of topography. This is a more rigorous scientific method than testing a fixed suite of processes against a fixed set of observables in order to decide wherein lies the ultimate truth. In the present work, the following are compared: diffusive hillslope processes, fluvial erosion and deep-seated bedrock landslides. In the remainder of this chapter, the methods used to represent these processes will be explained.

### 5.3 Framework for Landscape Modelling

Two common frameworks for long-term landform evolution modelling are discrete and continuous models<sup>1</sup>. An example of a discrete model is a cellular automaton, in which a set of rules is applied to cells in a (usually regular) grid (Bak *et al.*, 1987; Chase, 1994; Rodríguez-Iturbe and Rinaldo, 1997). In an atomist world discrete models ought to rule, however continuous models are often applied in situations where particles or

---

<sup>1</sup>Naturally, all numerical models must be solved in discrete space and time if a (conventional) computer is used; the distinction at issue here is the framework in which the model is formulated.

quanta are very small compared to the spatial scales of interest. Owing to the large size difference between landscape particles and river basins, a continuum approach will be taken here; it is described below<sup>2</sup>.

### 5.3.1 Continuum Model of Landscape Evolution

The most axiomatic step in the construction of a continuum model of landscape evolution is to consider elevation as a scalar field  $z(x, y)$ , where  $z$  is elevation and  $x, y$  are mutually-orthogonal horizontal space variables. Following this initial step, it is possible to derive a transport theorem for elevation (as if it were energy, mass or momentum), based on the assumption that volume is conserved<sup>3</sup>. The local form of this (Eq. 5.1) balances the change in elevation over time at a point,  $(\partial z/\partial t)$ , with the introduction to (or removal from) the model domain of landscape material,  $\Phi$ , and the flux of sediment from that point,  $(\mathbf{q} = -K(z, z_{x_i}, t)\partial z/\partial x_i)$ :

$$\frac{\partial z}{\partial t} = \frac{\partial}{\partial x_i} \left[ -K(z, z_{x_i}, t) \frac{\partial z}{\partial x_i} \right] + \Phi_U + \Phi_F + \eta(z, z_{x_i}) \quad (5.1)$$

where  $t$  = time;  $x_i$  = space variables;  $K$  = diffusivity function; and  $z_{x_i} = \partial z/\partial x_i$ . The production function,  $\Phi$ , is divided into three parts, each to represent a different landscape process:  $\Phi_U$  represents uplift,  $\Phi_F$  represents fluvial erosion, and  $\eta(z, z_{x_i})$  represents stochastic deep-seated bedrock landslides.

---

<sup>2</sup>The distinction between discrete and continuous models is therefore one of convenience: Gershfeld (1999) demonstrates analytically (using statistical mechanics) that a gas-lattice cellular automaton (discrete) exhibits exactly the behaviour of a fluid obeying the Navier-Stokes equations (continuous). A statistical approach is interesting in the context of modelling sediment transport in geomorphology because, although this is an area where particles are often much smaller than the space-scales of ultimate interest, the results of non-trivial interactions between these particles has been difficult to capture using traditional continuum mechanics.

<sup>3</sup>See Appendix B for a full derivation of this and subsequent results stated in this section. The assumption of conservation of volume is necessary to recover landscape form. Over longer timescales, it is more appropriate to enforce conservation of mass, however landscape form would not easily be modelled using this approach. Since morphology is central to the present work, a volume-conservative approach is more suitable here.

The individual components of Equation 5.1 are discussed below.

**Diffusion** The first term represents the flux of sediment away from the point: it is a diffusive term and can be used to account for processes that can be averaged over the time-scales of interest. These include creep, shallow landsliding and debris flows (see Chapter Three for justification of diffusion in the present study area). Physical intuition suggests that the rate of sediment transport by these processes should be related to slope, since they are driven primarily by gravity. The diffusivity function can be a constant, or a function of space and time variables and their derivatives. Empirical work by Martin and Church (1997) and Martin (1998, 2000) has shown that a suitable relation between slope and sediment flux is a non-linear sigmoidal curve (see Chapter Three). This is intuitively appealing because it suggests that, while sediment flux is low at low gradients, it increases rapidly until slopes reach approximately 45 degrees. Above this angle, failure is through different mechanisms that cannot be approximated by diffusion over geomorphic timescales. Martin (2000) has shown that an hyperbolic tangent function gives a suitable fit, which she has parameterized using data from the Queen Charlotte Islands. Details of a parameterization of the same function for the GVRD Capilano watershed are given in Chapter Three.

**Uplift** Tectonic and isostatic uplift may be included as a production term. In the present study uplift is included only in long model runs; uplift rates are poorly constrained compared with the detailed denudation observations used in this work, therefore in shorter runs ( $\sim 10$  ka) the effects of uplift are not studied.

**Fluvial Erosion** Fluvial erosion and deposition are included in the model based on a functional relation between slope,  $S$ , and stream discharge,  $Q$ , obtained by

simplifying the Bagnold equation presented by Martin and Church (2000). The simplified version, derived in Appendix B, is:

$$\Phi_F = \begin{cases} 6.44 Q^{0.35} S^{1.5} & \text{if } S < 0.01 \\ 40.32 Q^{0.34} S^{2.22} & \text{if } S > 0.01 \end{cases} \quad (5.2)$$

Following Martin (1998), a correction is made to account for configurational stability in steep headwater channels (i.e., those with  $S > 0.01$ ); ordinary fluvial processes are assumed not to operate where  $S > 0.1$ . At regular intervals the drainage network is rerouted using the D8 flow-routing algorithm (Jenson and Domingue, 1988).

Discharge is computed from upslope area,  $A$ , using the results of Church (1997):

$$Q = k_{fl} A^{0.68} \quad (5.3)$$

where  $k_{fl}$  is a parameter to represent hydroclimate.

**Deep-Seated Landslides** In order to incorporate the largest landslides, the effects of which cannot be averaged over the time-scales of interest, the stochastic term  $\eta(z, z_{x_i})$  is added.

The use of a stochastic term to represent large landslides is justified on the basis that the precise location of these events cannot be determined from variables resolved in the model. Large, deep-seated bedrock landslides occur as a result of a combination of weathering, cumulative hydrometeorologic conditions and seismic vibration; their location is determined by the distribution of these forcings in space and time. It is not practical to resolve these forcings over the time-scales of the present study. In addition, the exact location of rock slides is contingent

on local features of the bedrock configuration, such as bedding planes and the stratigraphic sequence, which are not codified or analysed. Despite these claims, it is possible to constrain the areas most likely to fail on the basis of local slope, relief and elevation. Consequently, in the present scheme each node is assigned a probability of failure,  $P_{sl}$ , based on local slope, relief and elevation criteria. At regular intervals,  $I_{sl}$ , model failure sites are chosen at random from a subset of model nodes where  $P_{sl}$  exceeds a critical value,  $P_{crit}$ . The failed material is then routed downslope in the direction of steepest descent so that its angle of repose is  $\phi_t$ .

The stochastic term makes Equation 5.1 a non-linear Langevin equation similar to the type that Sornette and Zhang (1993) discuss. This type of equation is well known to create surfaces with interesting fractal properties. Rodríguez-Iturbe and Rinaldo (1997) suggest a similar formulation, but derive the stochastic forcing from hydrological parameters rather than from hillslope-process measurements.

### 5.3.2 Parameters

As stated in Equation 5.1, the model involves several parameters. These are listed in Table 5.1, along with methods for their estimation from empirical data, and order of magnitude estimates (exact values used in model runs are given in Section 5.4).

Parameter	Symbol	Estimate	Source of Data
Diffusivity (average value)	$K$	$0.2 \text{ m}^2 \text{ a}^{-1}$	Volumetric slide measurements from air photos and field checking (Rood, 1984, 1990; Martin, 1998; GVRD, 1999)
Uplift	$U$	$0.002 \text{ m a}^{-1}$	Fission track dating of uplifted rocks (Parrish, 1983)
Hydroclimate	$k_A$	0.15	Analysis of hydrological records (Church, 1997)
Channel Support Area	$A_{\text{crit}}$	$2000 \text{ m}^2$	BC TRIM DEMs
Critical Slide Probability	$P_{\text{crit}}$	1.6 rad	Slide inventory (Cruden, 1985)
Minimum Slide Interval	$I_{\text{sl}}$	100 a	Slide inventory (Hungr <i>et al.</i> , 1999)
Angle of Repose (slide debris)	$\phi_t$	30 deg	(Middleton and Wilcock, 1994)

Table 5.1: Parameter Estimates for Landscape Evolution Model

### 5.3.3 Canonical Scaling

Using the order-of-magnitude estimates given in Table 5.1, and the structure of the model in Equation 5.1 it is possible to arrive at preliminary conclusions regarding the relative importance of the components of the model by making the individual terms of Equation 5.1 dimensionless. This is achieved with the following scalings:  $z = [z]z^*$ ,  $t = [t]t^*$ ,  $x_i = [x_i]x_i^*$ ,  $K = [K]K^*$ ,  $\Phi_U = [\Phi_U]\Phi_U^*$ ,  $\Phi_F = [\Phi_F]\Phi_F^*$ ,  $\eta = [\eta]\eta^*$ . Applying these to Equation 5.1 gives:

$$\frac{\partial z^*}{\partial t^*} = \frac{[t][K]}{[x_i]^2} \frac{\partial}{\partial x_i^*} (K^* \frac{\partial z^*}{\partial x_i^*}) + \frac{[t][\Phi_U]}{[z]} \Phi_U^* + \frac{[t][\Phi_F]}{[z]} \Phi_F^* + \frac{[t][\eta]}{[z]} \eta^* \quad (5.4)$$

from which the natural time constant for diffusion can be obtained. The time (in years) for elevation perturbations in a 1 km<sup>2</sup> area to become completely flat, assuming that only diffusive processes are operating, is:

$$[t] = \frac{[x_i]^2}{[K]} = \frac{(1 \times 10^3)^2}{0.02} = 5 \times 10^6 \text{ a} \quad (5.5)$$

Further results can be derived from Equation 5.4, including an expression that compares the relative roles of uplift and diffusion using the quotient of their respective characteristic scales:

$$\text{Uplift Ratio} = \frac{[\Phi_U][x_i]^2}{[z][K]} = \frac{(0.002)(1000)^2}{(1000)(0.2)} = 10 \quad (5.6)$$

In this case, the relative importance of uplift is greater than 1.0, indicating that landscapes in the present study-area are dominated by uplift, which is consistent with their tectonic setting and current relief.

### 5.3.4 Boundary Conditions

Conditions must be specified for the geometric boundaries of the model. Typical choices include constant-value boundaries, constant-derivative boundaries and combinations of these. Since neither fixed elevation nor fixed flux is a physically correct description of the situation a more complex set of boundary conditions is required. The model will be run for individual watersheds, so the physical feature at the boundary will be a ridge. The most realistic model boundary may therefore be a "reflection" of the topography within the model domain. Using the method of images to produce a reflected ridge is an appropriate approach; it is also necessary to include a larger area of DEM than is of interest (i.e., to include a buffer zone within which results are computed but then discarded). In this way, the natural structure of the landscape isolates the area of interest from spurious effects at the model edge.

### 5.3.5 Numerical Schemes

For simple cases with only linear hillslope diffusion, the problem can be solved analytically, yielding equilibrium solutions (e.g., Kirkby, 1971). Owing to the complex topography of a real drainage basin, a numerical approach will be taken here.

The choice of numerical schemes is first between finite difference and finite element models. The former discretize the governing equation on a regular grid; the latter use non-rectangular facets (commonly but not necessarily triangles). The advantages and disadvantages of each method are similar to those between lattice and triangular irregular network (TIN) representations of digital elevation data (see Kummer, 1994, for a recent comparison). In general, lattices (finite difference schemes) are simpler and easier to manipulate, however TINs (finite element schemes) are able to model complex surfaces more economically. The TIN structure is also appropriate for

modelling with different resolutions on hillslopes and channels. However, algorithms for flow-routing on TINs are in their infancy, which make them less suitable for simple models at present (McAllister and Snoeyink 1999; McAllister pers. comm., 1999). Consequently, a finite difference scheme was employed for this work.

### 5.3.6 Stability

A further choice is between explicit and implicit solutions on the finite difference grid. The conditionally-stable explicit solution takes the form of a Taylor series expansion of the differential equation resulting in an equation for the height of each node in the domain at time  $t + 1$ , which may be written solely in terms of the values of its neighbouring nodes at the earlier time,  $t$ .

Conversely, the implicit method produces an equation at each node for time  $t + 1$ , which is in terms of the value of that node at time  $t$ , and at several neighbouring nodes at time  $t + 1$ . Since, in this case, the future value of each node depends on the future value of every other node, the implicit method requires the solution of a set of simultaneous algebraic equations. Imaginably, this method is more computationally demanding especially for the non-linear case, but it is unconditionally stable.

Owing to its simplicity, an explicit solution is used here. In order to achieve this, it is necessary to use a forward-difference approximation for the time-derivative at the same time as a central-difference approximation for the spatial derivatives. This introduces errors of different orders of magnitude, and a problem of instability arises if the model is advanced for large values of  $\Delta t$  (although the solution is accurate for small values of  $\Delta t$ ).

The largest time-step for which the model can be run is therefore inversely proportional to the square of the chosen space step. Typically this time is much

smaller than the time constant for the problem (which is the time for any initial structure to be made uniform). Happily, in the present context, one is content to stop the model before the topography has disappeared, so one can tolerate the small time-step that an explicit scheme requires, particularly since the approach is very fast and a simple modification can take care of the non-linear case. The final C++ code is in Appendix B.

## 5.4 Results: Preliminary Runs

Preliminary runs were conducted to assess the initial plausibility of the model. Results from the most comprehensive run are shown in Figure 5.1, which is the final surface resulting from the evolution of a triangular prism (to represent a mountain ridge) with diffusion, fluvial processes and deep-seated bedrock landsliding over 100 ka. The initial surface was planar with no perturbations: all structure was derived from the erosion rules<sup>4</sup>.

The results of this run show the emergence of linear channels, characteristic of the present study area (and others with high uplift ratios; e.g., parts of New Zealand, Taiwan and other Pacific Islands). In addition, 1,000 deep-seated bedrock landslides occurred over the 100 ka period. These have altered the drainage network across the model domain, but particularly in the central upland portion where significant tributaries have been created.

An ensemble of 30 model runs was made, yielding an average magnitude-frequency distribution of rock slides, and allowing an estimate of its variability to be obtained. This distribution, which is shown in Figure 5.2, spans approximately 2

---

<sup>4</sup>This example is not intended to reproduce the exact topography in any natural location, since few landscapes begin as plane-faced triangular prisms, and it is not reasonable to expect that climate parameters or uplift rates will have remained the same throughout the 100 ka time period; they will almost certainly have changed.

orders of magnitude, however it does not follow a power-law. Figure 5.2 is similar to the distribution of the GVRD surficial slides (Fig. 3.1), showing a low-magnitude kink and a rounded central portion. Two suggestions follow from this: that the lower-end kink is an emergent feature of threshold exceedence failures, rather than an observational inadequacy; and that the distribution of modelled rock slides is exponential rather than power-law.

The observations presented for this model-run differ from the findings reported by Hergarten and Neugebauer (1999) in their simpler (but similar) model, and the experimental findings of Densmore *et al.* (1997). In these cases, the observation of power-law distributed rock slides has been used to suggest that landslides are a results of self-organized critical behaviour. An alternative explanation is suggested here.

In the present case, the rock slide interval was seen to have the most important effect on the resulting rockslide distribution. A further model ensemble was gathered with a higher rock-slide interval of 400 years. This group (Figure 5.3) has a magnitude-frequency distribution that more closely resembles a power-law (though it is not perfect). The effect of an increased slide interval is to reduce the total number of slides that occur, and to allow slopes to become steeper before they are likely to fail. The result of these differences is a reduction in the proportion of average-sized events, leading to a more extreme distribution.

It is, at present, unclear whether the power-law-like distribution results from the limitation imposed on the number of rock slides in the latter run (Figure 5.3, and cf. the sampling limitation imposed on a human observer), or whether it is a result of an increase in the degree to which slopes may be steepened between slides. The latter situation is important in the present regional context since it implies that oversteepening introduced by rapid tectonic uplift and as part of the legacy of glaciation is a necessary condition for the observation of power-law (or perhaps exponentially)

distributed slope-failures. In the search for answers to the questions raised so far, it is anticipated that a modelling approach may prove useful.

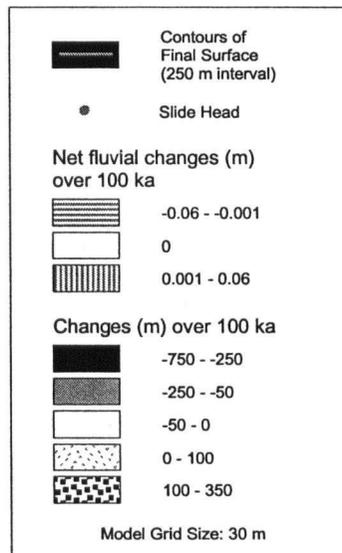
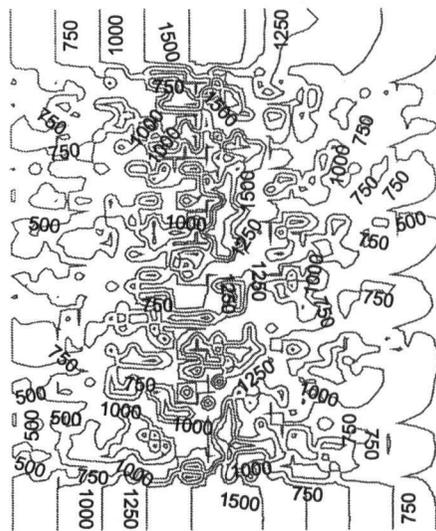
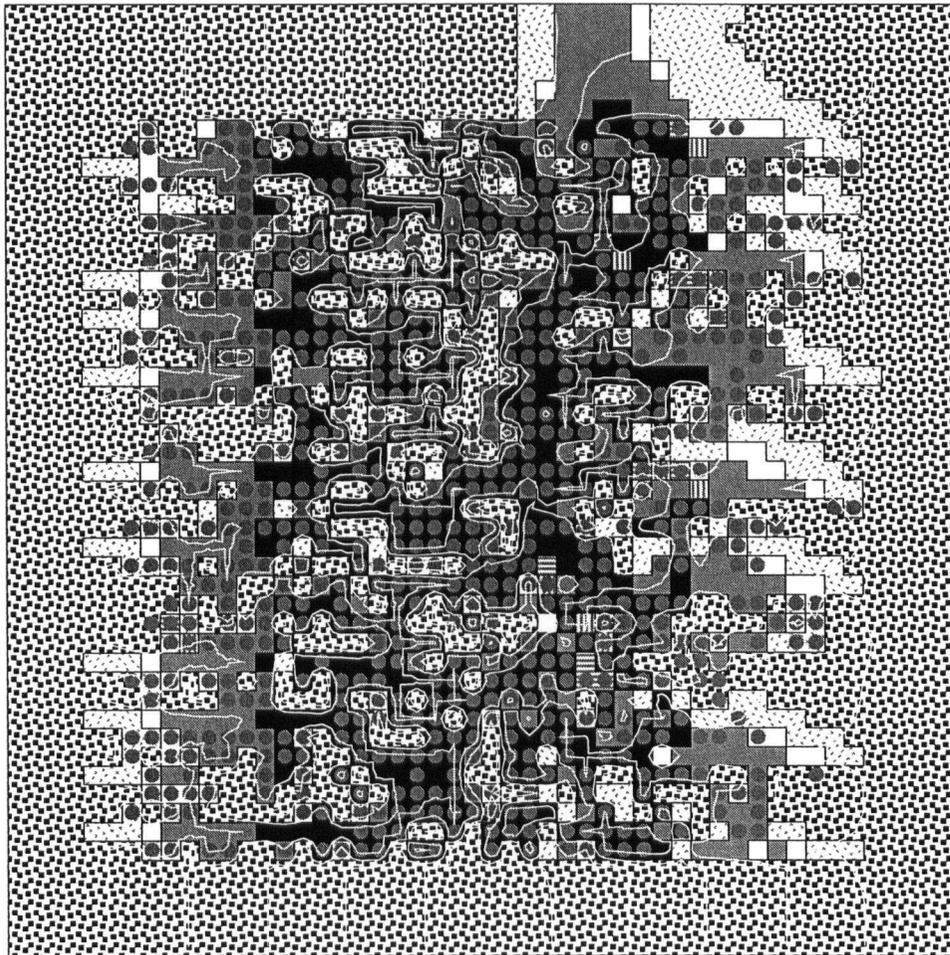


Figure 5.1: Artificial Model Surface after 100ka

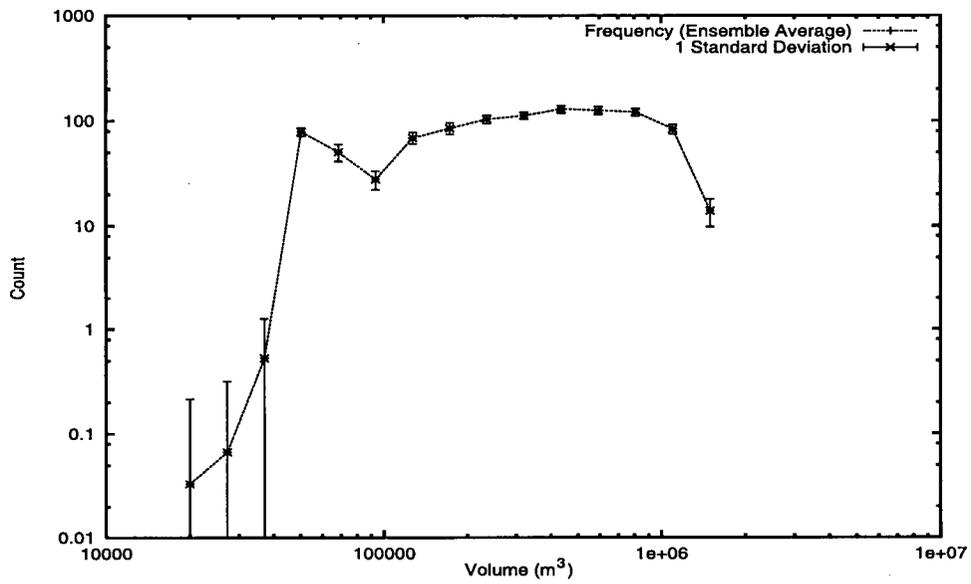


Figure 5.2: Model Rock Slides: Magnitude-Frequency (100 year slide interval; N = 1000)

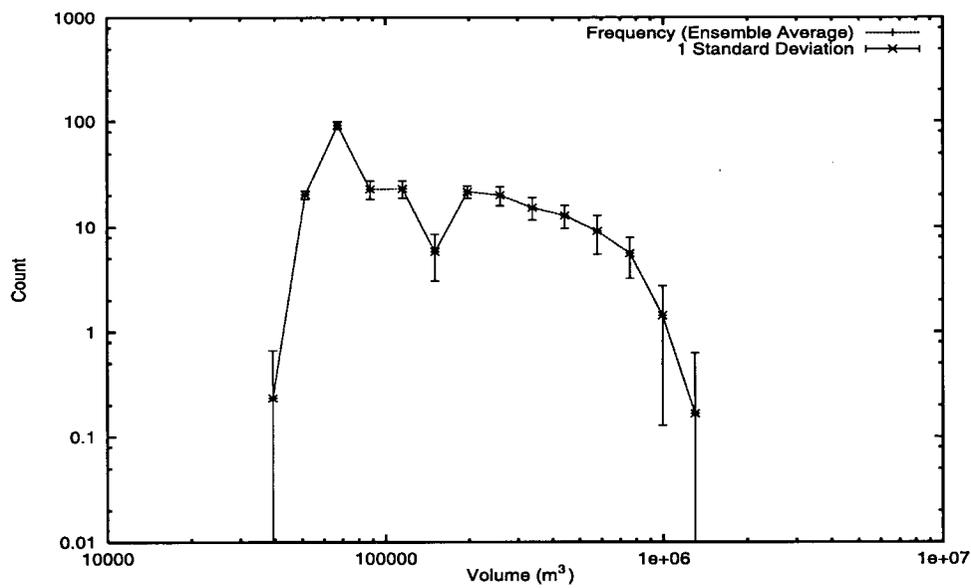


Figure 5.3: Model Rock Slides: Magnitude-Frequency (400 year slide interval; N = 250)

## 5.5 Results: Detailed Model Runs

A more realistic use of the landform evolution model presented here is to examine the predicted changes over a shorter time period for a real landscape. In this section, the results of such an application are discussed, and the fractal dimensions of various hypothetical process-domains are presented in order to approach an answer the previously posed questions regarding the effect of episodic processes such as deep-seated bedrock landslides on the fractal dimension of topography.

One sub-basin was chosen within each of the study areas: Lembke Creek, a 3.6 km<sup>2</sup> sub-basin in the western side of the Capilano watershed (see Fig. 2.3); and a 4.4 km<sup>2</sup> tributary of Gregory Creek, which is in the Rennel Sound area of the Queen Charlotte Islands (Fig. 2.2). Small basins were chosen in order to reduce topographic complexity and to preserve computational tractability. Table 5.2 gives the parameters used in each model run, the details of which are discussed below. All runs were for 10 ka, using the appropriate non-linear diffusion rule (see Chapter Three) on a regular square grid of 30 m on a side. The effects of tectonics were not included, since they obscure the subtleties of other processes.

Watershed	Hydroclimate Church (1997)	Hillslope-Fluvial Threshold ( $m^2$ ) (see Chapter Four)	Rock Slide Threshold (slope * elevation) $m$ (see Chapter Four)
Lembke	0.15	2000	1000
Gregory	0.17	6000	450

Table 5.2: Parameters Used in Model Runs

### 5.5.1 Lembke Creek

Results for Lembke Creek without deep-seated rock slides are shown in Figure 5.4. This map shows contours of the final surface. Changes between the start and finish of the model run are shown in reds (erosion) and greens (deposition)<sup>5</sup>. Areas of net fluvial erosion are shown in blue; net fluvial deposition is marked in grey (N.B.: values for fluvial changes have been divided over the whole cell in which the river lies (900 m<sup>2</sup>)). The steep slopes to the north-west have yielded considerable amounts of sediment that has been transported onto the valley floor by headwater streams. Most of this sediment is from areas where slopes have been over-steepened by glaciation or uplift. The effect of fluvial processes is to homogenize the valley flat, by redistributing sediment that is supplied from the hillsides. The area to the south-east shows a lower amount of erosion by diffusive processes: it is a steep rock slope and therefore reasonably stable in this model run.

Figure 5.5 shows the same situation as the previous run, but with the addition of stochastic deep-seated bedrock landslides. Slide heads are shown in yellow, and tracks are clearly visible as areas of high erosion. Deep-seated slides are common in the south-eastern rock slopes, where they have altered the drainage network and the rock face. Areas of deposition on lower slopes beneath large landslides (shown in green) may be interpreted as talus cones (or alluvial fans in the isolated cases along the central and eastern slopes of the southern valley wall).

The interaction of channel and hillslope processes is especially noteworthy: channel processes are able to respond to changes in the topography induced by hillslope processes (episodic and otherwise) because the channel network is recomputed from the topography after each rock slide (every 100 years). The effect of this is to

---

<sup>5</sup>Shading and symbols are used in place of colour on black-and-white archival versions; in each case the appropriate key is given on the figure.

re-route flow into the crenulations created by rock slides, and to allow the passage of colluvial deposits into river channels. Examples of the former can be seen along the southern valley wall; of the latter on the most southerly talus cone.

The primary reason for having created this model is to assess the effect of the erosion rules on the fractal dimension of the topography. Table 5.3 shows the fractal dimensions of the channel networks at 2,500 year intervals under the influence of diffusion and fluvial processes and with the addition of rock slides. It also shows the division of the non-fluvial part of the landscape into hypothetical process-domains (upland areas with upslope area 0 – 1,800 m<sup>2</sup>; and lowland areas with upslope area 1,800 – 5,400 m<sup>2</sup>). The procedures used for the estimation of these parameters were identical to those used in Chapter 4 for the original DEM data.

Time <i>a</i>	$D_{channel}$	$D_{upland}$	$D_{lowland}$
With Rock Slides			
0	1.62	2.17	2.21
2500	1.64	2.14	2.19
5000	1.46	2.13	2.19
7500	1.46	2.13	2.19
10000	1.50	2.14	2.19
No Rock Slides			
0	1.62	2.17	2.21
2500	1.74	2.15	2.20
5000	1.69	2.15	2.20
7500	1.71	2.15	2.20
10000	1.58	2.15	2.20

Table 5.3: Fractal Dimensions of Model Results

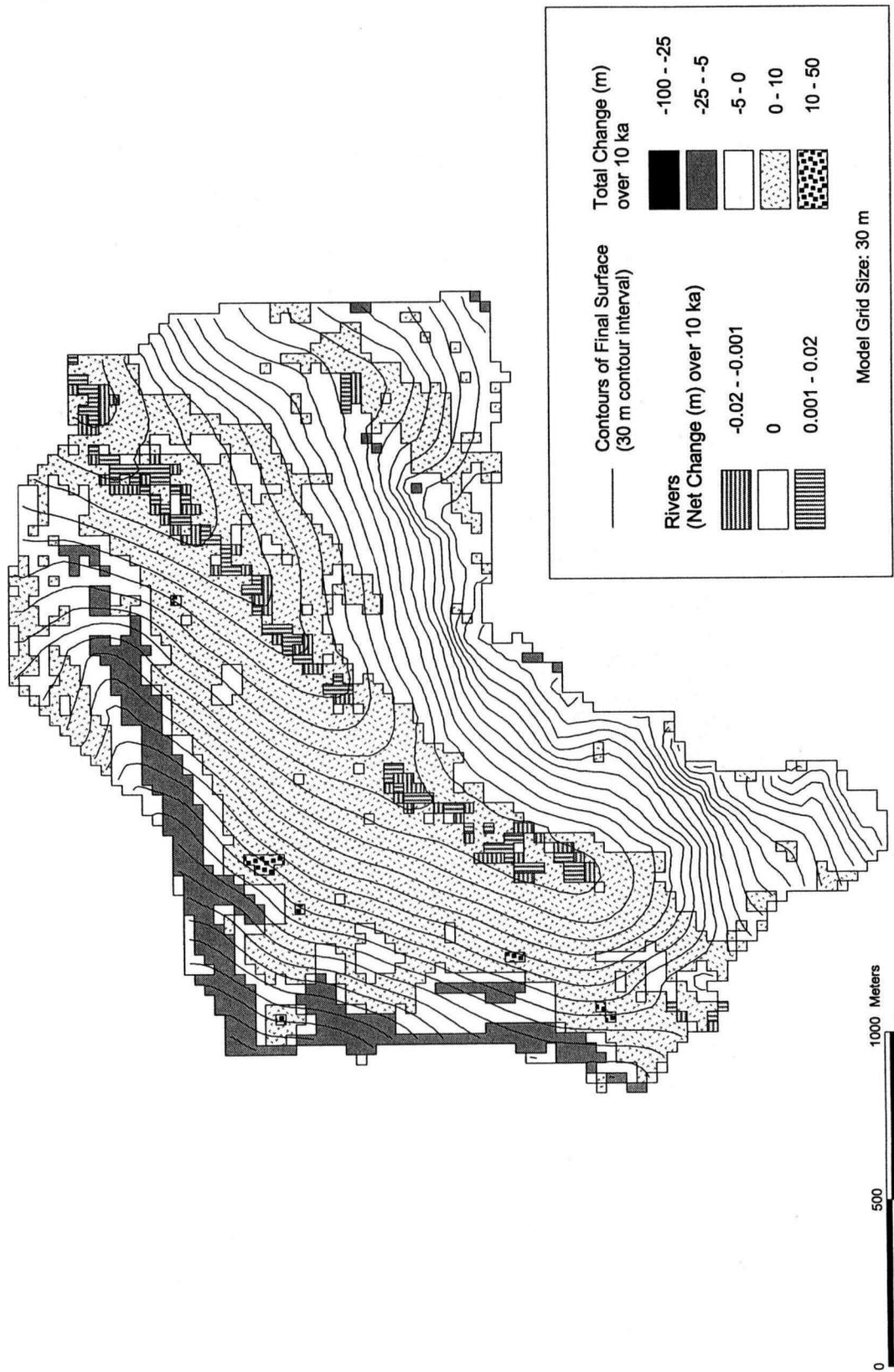


Figure 5.4: Lembke Creek: 10 ka no rock slides

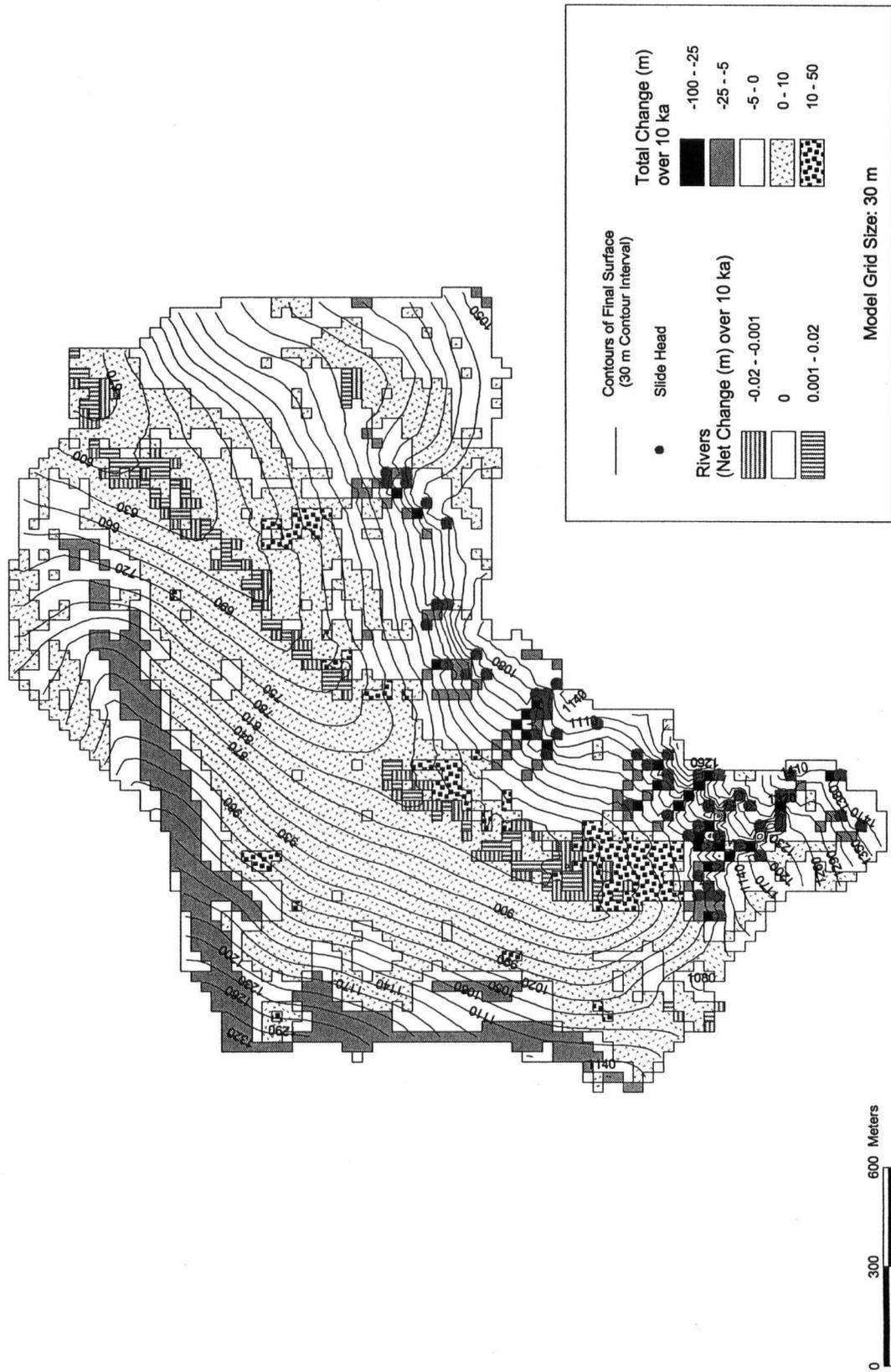


Figure 5.5: Lembke Creek: 10 ka with rock slides

The fractal dimension of the channel network responds as expected: in the case without rock slides, the channel network fractal dimension fluctuates about an average value of 1.67. The addition of rock slides liberates large quantities of material from rock slopes, filling the channel network and restricting the areas where evidence of fluvial processes may be found. This leads to a systematic decrease of the fractal dimension of the channel network over the course of 10 ka.

The effect of hillslope processes is less obvious. Of particular note is that the upper hillslopes in each case have a *lower* fractal dimension than the lower hillslopes. This is the opposite of the result found in Chapter Four. This discrepancy may have resulted from the necessity to resample elevation data onto a coarser grid for modelling work in order to relieve computational constraints (this was achieved using ARC/INFOs cubic interpolation algorithm). The effects of DEM resolution on the determination of fractal properties of landscape have not, to my knowledge, been examined in the literature.

The effect of diffusion, which can be seen most clearly in Figure 5.4, is to smooth the surface; this is captured in the fractal dimension, which declines from 2.17 to 2.15 (upland) and from 2.21 to 2.20 (lowland) over the 10 ka period. The change may be smaller than the assumed precision of the technique for estimating  $D$ ; nevertheless, the decline in  $D$  is consistent.

The addition of bedrock landslides has a roughening effect visible in Figure 5.5, however they cause a *decrease* in the fractal dimension, indicating a relative smoothing effect. There may be several explanations for this result: (i) bedrock landsliding occurs over only a limited range of scales and therefore destroys part of the fractal scaling introduced by fluvial processes; or (ii) the traces of landslides quickly become part of the channel network and therefore are excluded from the fractal analysis presented here (using the method of Veneziano and Iacobellis, 1999). The first explanation is

consistent with the idea of pseudo-fractal topography suggested by Burrough (1993); it implies that, while individual landslides decrease the fractal dimension, their cumulative interference may lead to an increase in the fractal dimension over much longer time-scales; 10 ka is therefore not sufficient to create pseudo-fractal landscapes.

The outright acceptance of the second explanation is difficult because, if it were true, one would expect deep-seated landslides to contribute to an increased channel network fractal dimension, which is not the case. A third explanation might therefore be that the deep-seated landslides create a different type of sediment flow-path that is missed by the analysis presented here. The landslide flow paths have a higher slope than would be expected if they were fluvial channels, but still have sufficiently high upslope areas that they are influenced by the flow of water. These configurationally-stable channels on steep slopes do not fit well into the distinction between hillslope and fluvial processes, and further work is necessary to enable these hypotheses to be tested. The first steps are likely to involve better representation of channel processes on slopes greater than 0.1, since they were not considered in the present study.

### 5.5.2 Gregory Creek

Results from Gregory Creek are shown in Figure 5.6. The most notable feature here is that although the landscape was tested for areas likely to slide with the same regularity as the model of Lembke Creek (above), the occurrence of deep-seated landslides was confined to the most southerly area where two small rock slides occurred; they were not large enough to create a contour crenulation. This was found to be the case for all physically realistic values of the rock slide stability threshold, suggesting that there is an underlying topographic reason for the lack of large rock slides in Gregory Creek. This conjecture is supported by the evidence of Ryder (pers. comm., 2000), whose

recent aerial-photograph inventories of slope processes on the Queen Charlotte Islands show little evidence of large deep-seated landslides. Most slope failures in Gregory Creek are shallow landslides in surficial material: for long-term landform modelling in this area non-linear diffusion alone can account for all sediment flux from hillslopes.

The fluvial network shown in the results from Gregory Creek contains a greater amount of deposition than that for Lembke Creek. In particular there is a large area of deposition at the confluence of several streams as they emerge from the upland area in the eastern part of the map. The reproduction of this feature, which represents an alluvial fan, gives confidence in the way that fluvial processes have been incorporated into the model.

The fractal properties of this Gregory Creek tributary were not studied because the rarity of rock slides means that there is no basis for comparison between models with and without them.

## 5.6 Discussion

The results presented in this chapter show that diffusive hillslope processes lead to a reduction in the fractal dimension of topography. In cases where rock slides do not occur, the fluvial network removes material in such a way as to preserve the fractal dimension of the channel network.

The addition of episodic events has an effect that is more complex than simply roughening the surface (although roughening is clearly observed in Figure 5.5). Indeed, the fractal dimensions of hillslopes in model runs with structure generating landslides were *lower* than those of hillslopes without these perturbations.

Two possible reasons for this result are suggested. The first is that while slope failures often appear to be power-law distributed, this holds over only a limited range

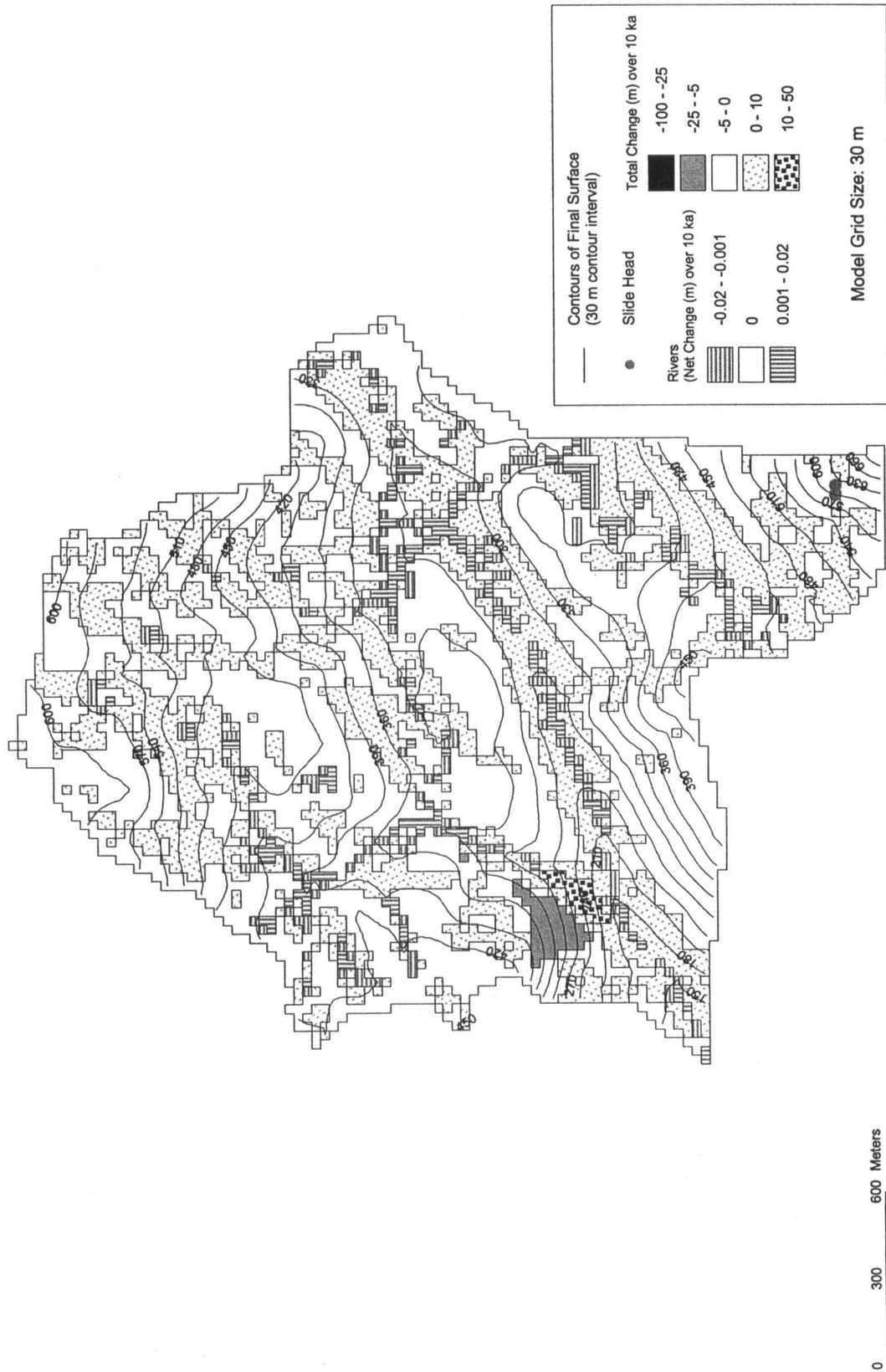


Figure 5.6: Gregory Creek: 10 ka with rock slides

of scales; therefore rockslides *do* have a characteristic range of length scales and their occurrence leads to a reduction in the fractal dimension of parts of the landscape in which they occur. This may, over long time periods, create a pseudo-fractal landscape of the type described by Burrough (1993).

An alternative explanation is that rock slides quickly modify the drainage network by creating steep headwater channels or gullies, which are not identified on slope-area plots (even using the procedure of Veneziano and Iacobellis) because they are too steep to be channels, yet have too large a contributing area to be hillslopes. The solution of this problem is non-trivial and would require a study of the morphometry of headward channels. Further work on hillslope-fluvial process coupling is also necessary in order to decide between the these two explanations. This has also been suggested by Martin (1998), Hovius *et al.* (2000) and Beaumont *et al.* (2000), owing to the realization of the importance of headwater channels in long-term landscape evolution, which is summarized in Church (in press).

## Chapter 6

### Conclusion

The work in this thesis has attempted to test two hypotheses concerning geomorphic processes and the fractal properties of topography: (i) that there exist areas of the landscape with distinct fractal dimensions; and (ii) that these fractal domains are created by the operation of characteristic geomorphic processes.

Several conclusions can be made. In Chapter Three, landslides on the Queen Charlotte Islands were shown to have a power-law magnitude-frequency distribution similar to those observed in other regions by Hovius *et al.* (1997), Pelletier *et al.* (1997) and Turcotte (1999). The slope-volume relation for a landslide inventory in the GVRD Capilano River basin was shown to be non-linear, in agreement with the hyperbolic tangent relation found by Martin (1998, 2000) for the Queen Charlotte Islands landslides. In Chapter Four, it was shown that evidence of power-law landslide statistics may be reflected in the non-integer fractal dimension observed for hillslopes in both the GVRD and Queen Charlotte Islands watersheds. Using a modified version of Veneziano and Iacobellis' (1999) procedure for separating areas dominated by hillslope and channel processes, it was shown that, in several cases, hillslope process-domains have a higher fractal dimension. In all cases, there is a consistent difference between the fractal dimension of topography in the hillslope and fluvial domains, supporting

the first hypothesis that process domains have distinct fractal dimensions. It is further suggested that details of the spatial variability of the fractal dimension depend on the glacial history of individual areas.

The second hypothesis, that spatial differences in fractal dimension can be related to geomorphic processes, was examined in Chapter Five. Results from a landscape evolution model show that smoothing by diffusion is observed in the fractal dimension of simulated topography. Large episodic landslides are shown to have a short-term impact on the perceived “roughness” of the surface, however the perturbations that they create are quickly incorporated into the drainage network, where they appear as steep headwater channels (which are a common feature of landscapes in the area of study). The influence of deep-seated bedrock landslides on the fractal dimension of topography remains unclear: landslides reduce the fractal dimension in the short term, but their cumulative effect may be to increase it over longer periods of time, creating a pseudo-fractal landscape. In conclusion, the second hypothesis requires further work before it can be fully accepted. It is suggested that a numerical modelling approach is appropriate, but that further attention should be given to geomorphic processes in mountain streams.

## **6.1 Practical Implications**

The practical utility of the findings reported here is in the area of long-term planning for land-use management and risk mitigation. The magnitude-frequency distributions presented for slope failure may be used to assess the likelihood of failure of a given volume over a representative time period. The landform modelling approach used in Chapter Five is also of relevance to land-use planning: its results are not deterministic predictions of landscape change; rather, they indicate areas of potential long-term

instability and allow the identification of important fluvial and hillslope sediment pathways. At time-scales longer than 100 years deterministic predictions are of limited use: generalized probabilistic predictions are more informative, and of greater utility in applications in which a measure of the uncertainty of predicted events is required. A probabilistic approach is especially useful in view of non-steady-state variability in erosion rates over long timescales (cf. Benda, 1994). In this respect, the ensemble approach taken in Chapter Five is useful: the model results shown there are generalized probabilistic predictions, with an attached measure of uncertainty.

An important line of further work in the use of the current approach is to assess the ways in which modification by human activity can be included. Initially, this might include modification of slope-failure rates resulting from timber harvesting or other changes from natural land use. Modification of fluvial processes in response to climate changes is also possible, although the correct calibration of such scenarios may present difficulty.

## 6.2 Theoretical Implications

The theoretical implications of the work presented here concern the use of fractal properties of topography to understand geomorphic processes. It has been confirmed that splitting landscapes into fluvial and hillslope process-domains can lead to the creation of areas with homogeneous fractal dimension, however some landscape features do not fit well into the hillslope-fluvial distinction. These include headward channels and the remains of deep-seated bedrock landslides. It is also possible that landscape processes with a limited range of length scales can create non-fractal areas of landscape in the short term, yet over a long time their influence may lead to pseudo-fractal effects, as suggested by Burrough (1993).

A fundamental need in contemporary geomorphology is to relate fractal landforms to geomorphic processes. Several recent attempts have produced wide-reaching theories of drainage-basin-scale fluvial geomorphology (these include Rodríguez-Iturbe and Rinaldo, 1997; Veneziano and Iacobellis, 1999). Theories of long-term landscape evolution are essential to understanding the ways in which information is transferred across scales in the landscape. It is important that they include processes of regional significance, such as deep-seated bedrock landsliding, the controls of which are contingent on factors such as bedrock geology and landscape history.

## Bibliography

- Alley, N. F. and Thompson, B. (1978). *Aspects of Environmental Geology, Parts of Graham Island, Queen Charlotte Islands*. British Columbia Ministry of Environment, Resource Analysis Branch (Bulletin No. 2), Victoria, BC, Canada.
- Avnir, D., Biham, O., Lidar, D., and Malcai, O. (1998). Is the geometry of nature fractal? *Science*, **279**(5347), 39–40.
- Bak, P., Tang, C., and Wiesenfeld, K. (1987). Self-organized criticality: an explanation of  $1/f$  noise. *Physical Review Letters*, **59**, 381–384.
- Bak, P., Tang, C., and Wiesenfeld, K. (1988). Self-organized criticality. *Physical Review A*, **38**(1), 364–374.
- Barnsley, M. F. (1988). *Fractals Everywhere*. Academic Press, San Diego, CA.
- Beaumont, C., Kooi, H., and Willet, S. (2000). Coupled tectonic-surface process models with applications to rifted margins and collisional orogens. In M. A. Summerfield, editor, *Geomorphology and Global Tectonics*, pages 29–55. John Wiley & Sons Ltd., Chichester.
- Benda, L. E. (1994). *Stochastic Geomorphology in a Humid Mountain Landscape*. Ph.D. Thesis, Department of Geological Sciences, The University of Washington.

- Burrough, P. A. (1981). The fractal dimension of landscapes and other environmental data. *Nature*, **294**, 240–242.
- Burrough, P. A. (1993). Fractal and geostatistical methods in landscape studies. In N. S. N. Lam and L. de Cola, editors, *Fractals in Geography*, pages 87–121. Prentice Hall, Englewood Cliffs, New Jersey.
- Campbell, D. (1999). *Sediment Sources and Transfers in Lynn Valley, B.C.*. B. Sc. Honours Thesis, Department of Geography, The University of British Columbia.
- Chase, C. G. (1994). Fluvial landsculpting and the fractal dimension of topography. *Geomorphology*, **5**, 39–57.
- Church, M. (1997). *Regionalised Hydrological Estimates for British Columbia: First Approximation of Scale Effects*. Resource Inventory and Data Management Branch, British Columbia Ministry of Environment, Lands and Parks, Victoria, Canada.
- Church, M. (in press). Mountains and montane channels. In R. J. Allison and T. P. Burt, editors, *Sediment Cascades: An Integrated Approach*.
- Church, M. and Mark, D. M. (1980). On size and scale in geomorphology. *Progress in Physical Geography*, **4**, 342–390.
- Clague, J. J. (1989). Quaternary geology of the Canadian Cordillera. In R. J. Fulton, editor, *Quaternary Geology of Canada and Greenland*, pages 15–96. Geological Survey of Canada, Ottawa.
- Cruden, D. M. (1985). Rock slope movements in the Canadian Cordillera. *Canadian Geotechnical Journal*, **22**, 528–540.
- Culling, W. E. H. (1960). Analytical theory of erosion. *Journal of Geology*, **68**, 336–344.

- Culling, W. E. H. (1988). A unified theory of particulate flows in geomorphic settings. *Earth Surface Processes and Landforms*, **13**, 431–585.
- Davis, W. (1899). The geographical cycle. *Geographical Journal*, **14**, 481–504.
- Day, T. J. (1969). *The Channel Geometry of Mountain Streams*. M.Sc. thesis, The University of British Columbia.
- Densmore, A. L., Anderson, R. S., McAdoo, B. G., and Ellis, M. A. (1997). Hillslope evolution by bedrock landslides. *Science*, **275**(5298), 369–372.
- Duhem, P. (1905). *Physical Theory and Experiment*. Princeton University Press, Princeton, (trans. P. P. Weiner, 1954).
- English, R. R. (1996). *Lineament Control on Drainage Basin Development, Large Rock Landslides and Mountain Slope Deformation in the SW Coast Mountains, BC, Canada*. M.Sc. thesis, Department of Geological Sciences, The University of British Columbia.
- Evans, I. S. and McClean, C. J. (1995). The land surface is not a multifractal. *Zeitschrift für Geomorphologie N.F. Suppl.-Bd.*, **101**, 127–147.
- Gershenfeld, N. A. (1999). *The Nature of Mathematical Modelling*. Cambridge University Press, Cambridge.
- Gimbarzevsky, P. (1988). *Mass Wasting on the Queen Charlotte Islands (Land Management Report No. 29)*. British Columbia Ministry of Forests and Lands, Victoria, Canada.
- GVRD (1999). *Greater Vancouver Regional District Watershed Management Report No. 5*. Ecological Inventory Programme, Vancouver, Canada.

- Haines-Young, R. H. and Petch, J. R. (1986). *Physical Geography: its nature and methods*. Harper and Row, London.
- Hergarten, S. and Neugebauer, H. J. (1999). Self-organized criticality in landsliding processes. In S. Hergarten and H. J. Neugebauer, editors, *Process Modelling and Landform Evolution*, pages 231–251. Springer-Verlag, Berlin.
- Hewitt, K. (1989). The altitudinal organization of Karakoram geomorphic processes and depositional environments. *Zeitschrift für Geomorphologie N.F. Suppl.-Bd.*, **76**, 9–32.
- Horton, R. E. (1945). Erosional development of streams and their drainage basins: Hydrophysical approach to quantitative morphology. *Bulletin, Geological Society of America*, **56**, 275–370.
- Hovius, N., Stark, C. P., and Allen, P. A. (1997). Sediment flux from a mountain belt derived by landslide mapping. *Geology*, **25**(3), 231–234.
- Hovius, N., Stark, C. P., Tutton, M. A., and Abbot, L. D. (1998). Landslide-driven drainage network evolution in a pre-steady-state mountain belt: Finisterre Mountains, Papua New Guinea. *Geology*, **26**(12), 1071–1074.
- Hovius, N., Stark, C. P., Chu, H.-T., and Lin, J.-C. (2000). Supply and removal of sediment in a landslide-dominated mountain belt: Central Range, Taiwan. *Journal of Geology*, **108**, 73–89.
- Hungr, O., Evans, S. G., and Hazzard, J. (1999). Magnitude and frequency of rock falls and rock slides along the main transportation corridors of southwestern British Columbia. *Canadian Geotechnical Journal*, **36**(2), 224–238.

- Jenson, S. K. and Domingue, J. O. (1988). Extracting topographic structure from digital elevation data for geographic information system analysis. *Photogrammetric Engineering and Remote Sensing*, **54**(11), 1593–1600.
- Kadanoff, L. P. (1986). Fractals: Where's the physics? *Physics Today*, **February**, 6–7.
- Kirkby, M. J. (1971). Hillslope process-response models based on the continuity equation. *Transactions, Institute of British Geographers*, **Special Publication No. 3**.
- Klinkenberg, B. and Goodchild, M. F. (1992). The fractal properties of topography: A comparison of methods. *Earth Surface Processes and Landforms*, **17**, 217–234.
- Klinkenberg, B. and Goodchild, M. F. (1993). Statistics of channel networks on fractional Brownian surfaces. In N. S. N. Lam and L. de Cola, editors, *Fractals in Geography*. Prentice Hall, Englewood Cliffs, New Jersey.
- Knighton, D. (1998). *Fluvial Forms and Processes: A New Perspective*. Arnold, London.
- Kumler, M. P. (1994). An intensive comparison of triangular irregular networks (TINs) and digital elevation models (DEMs). *Cartographica*, **31**(2), 1–99.
- Lam, N. and de Cola, L. (1993). *Fractal in Geography*. Prentice Hall, Englewood Cliffs, New Jersey.
- Lavallée, D., Lovejoy, S., Schertzer, D., and Ladoy, P. (1993). Nonlinear variability of landscape topography: Multifractal analysis and simulation. In N. S. N. Lam and L. de Cola, editors, *Fractals in Geography*, pages 158–192. Prentice Hall, Englewood Cliffs, New Jersey.

- Leir, M. C. (1995). *Airborne Synthetic Aperture Radar, Digital Terrain Models and Geographic Information Systems: Tools for Mapping and Managing Large Land-slide Hazards in South-Western British Columbia*. M.A.Sc. thesis, Department of Geological Sciences, University of British Columbia.
- Lian, O. B. and Hickin, E. J. (1993). Late Pleistocene stratigraphy and chronology of lower Seymour Valley, southwestern British Columbia. *Canadian Journal of Earth Sciences*, **30**, 841–850.
- Mandelbrot, B. B. (1967). How long is the coast of Britain? statistical self-similarity and fractional dimension. *Science*, **156**, 636–638.
- Mandelbrot, B. B. (1983). *The Fractal Dimension of Topography*. Freeman, New York.
- Mark, D. M. and Aronson, P. B. (1984). Scale-dependent fractal dimensions of topographic surfaces: An empirical investigation with applications in geomorphology and computer mapping. *Mathematical Geology*, **16**(7), 671–683.
- Martin, Y. (1998). *Modelling Geomorphology in Long Term Landform Evolution*. Ph.D. thesis, Department of Geography, University of British Columbia.
- Martin, Y. (2000). Modelling hillslope evolution: Linear and non-linear transport relations. *Geomorphology*, in press.
- Martin, Y. and Church, M. (1997). Diffusion in landscape development models: On the nature of basic transport relations. *Earth Surface Processes and Landforms*, **22**, 273–279.
- Martin, Y. and Church, M. (2000). Re-examination of Bagnold's empirical bedload formulae. *Earth Surface Processes and Landforms*, **25**.

- McAllister, M. and Snoeyink, J. (1999). The geometry of watershed on TINs. Paper Presented at GIS'99, Vancouver, Canada.
- Mehta, A. (1992). Real sandpiles: Dilatancy, hysteresis and cooperative dynamics. *Physica A*, **186**, 121–153.
- Merritts, D. and Ellis, M. (1994). Introduction to special section of tectonics and topography. *Journal of Geophysical Research*, **99**(B6), 12,135–12,141.
- Middleton, G. V. and Wilcock, P. R. (1994). *Mechanics in the Earth and Environmental Sciences*. Cambridge University Press, Cambridge.
- Molnar, P. and England, P. (1990). Late Cenozoic uplift of mountain ranges and global climate change: chicken or egg. *Nature*, **346**, 29–34.
- Montgomery, D. R. and Foufoula-Georgiou, E. (1993). Channel network source representation using digital elevation models. *Water Resources Research*, **29**(12), 3925–3934.
- Nikora, V. (1991). Fractal structure of river planforms. *Water Resources Research*, **27**, 1327–1333.
- Oreskes, N., Shrader-Frechette, K., and Belitz, K. (1994). Verification, validation and confirmation of numerical models in the earth sciences. *Science*, **263**, 641–646.
- Outcalt, S. I. and Melton, M. A. (1992). Geomorphic applications of the Hausdorff-Besicovitch dimension. *Earth Surface Processes and Landforms*, **17**, 775–787.
- Parrish, R. R. (1983). Cenozoic thermal evolution and tectonics of the Coast Mountains of British Columbia: 1. fission track dating, apparent uplift rates and patterns of uplift. *Tectonics*, **2**(6), 601–631.

- Pelletier, J. D., Malamud, B. D., Blodgett, T., and Turcotte, D. L. (1997). Scale-invariance of soil moisture variability and its implications for the frequency-size distribution of landslides. *Engineering Geology*, **48**, 255–268.
- Penck, W. (1924). *Morphological Analysis of Landforms*. MacMillan, London, trans. H. Geck and K. C. Boswell, 1953.
- Quine, W. v. O. (1953). *From a Logical Point of View*. Harvard University Press, Cambridge, Mass.
- Rodríguez-Iturbe, I. and Rinaldo, A. (1997). *Fractal River Basins*. Cambridge University Press, Cambridge.
- Roering, J. J., Kirchner, J. W., and Dietrich, W. E. (1999). Evidence for non-linear, diffusive sediment transport on hillslopes and implications for landscape morphology. *Water Resources Research*, **35**(3), 853–870.
- Rood, K. M. (1984). *An Aerial Photograph Inventory of the Frequency and Yield of Mass-Wasting on the Queen Charlotte Islands, British Columbia (Land Management Report No. 34)*. British Columbia Ministry of Forests, Victoria, Canada.
- Rood, K. M. (1990). *Site Characteristics and Landsliding in Forested and Clearcut Terrain, Queen Charlotte Islands, British Columbia (Land Management Report No. 64)*. British Columbia Ministry of Forests, Victoria, Canada.
- Ryder, J. M. (1981). Geomorphology of the southern part of the Coast Mountains of British Columbia. *Zeitschrift für Geomorphologie N.F. Suppl.-Bd.*, **37**, 120–147.
- Savage, S. B. (1993). Mechanics of granular flows. In K. Hutter, editor, *Continuum Mechanics in Environmental Sciences and Geophysics*, pages 467–522. Springer-Verlag, Wien.

- Savigny, K. W. (1991). Engineering geology of large landslides in the Lower Fraser River valley transportation corridor southwestern Canadian Cordillera. In S. G. Evans, editor, *Landslide Hazards in the Canadian Cordillera*. Geological Association of Canada.
- Schumm, S. A. (1956). Evolution of drainage systems and slopes in badlands at Perth Amboy, New Jersey. *Bulletin, Geological Society of America*, **67**, 597–646.
- Shenker, O. R. (1994). Fractal geometry is not the geometry of nature. *Studies in the History and Philosophy of Science*, **25**(6), 967–981.
- Smith, R. B., Commandeur, P. R., and Ryan, M. W. (1986). *Soils, Vegetation and Forest Growth on Landslides and Surrounding Logged and Old-Growth Areas on the Queen Charlotte Islands (Land Management Report No. 41)*. British Columbia Ministry of Forests, Victoria, Canada.
- Snow, R. S. (1989). Fractal sinuosity of stream channels. *Pure and Applied Geophysics*, **131**, 99–109.
- Sornette, D. and Zhang, Y.-C. (1993). Non-linear Langevin model of geomorphic erosion processes. *Geophysical Journal International*, **113**, 382–386.
- Stølum, H.-H. (1996). River meandering as a self-organization process. *Science*, **271**, 1710–1713.
- Sutherland-Brown, A. (1968). *Geology of the Queen Charlotte Islands, British Columbia (Bulletin 54)*. British Columbia Department of Mines and Petroleum Resources, Victoria.
- Takahashi, T. (1981). Debris flow. *Annual Review of Fluid Mechanics*, **13**, 57–77.

- Tarboton, D. G., Bras, R. L., and Rodriguez-Iturbe, I. (1991). On the extraction of channel networks from digital elevation data. *Hydrological Processes*, **5**, 81–100.
- Tucker, G. E. and Bras, R. L. (1998). Hillslope processes, drainage density and landscape morphology. *Water Resources Research*, **34**(10), 2751–2764.
- Tucker, G. E. and Slingerland, R. L. (1994). Erosional dynamics, flexural isostasy and long-lived escarpments: A numerical modelling study. *Journal of Geophysical Research*, **99**(B6), 12,229 – 12,243.
- Turcotte, D. L. (1992). *Fractals and Chaos in Geology and Geophysics*. Cambridge University Press, Cambridge.
- Turcotte, D. L. (1998). *Fractals and Chaos in Geology and Geophysics*. Cambridge University Press, Cambridge (Second Edition).
- Turcotte, D. L. (1999). Self-organized criticality. *Reports on Progress in Physics*, **62**(10), 1377–1429.
- Veneziano, D. and Iacobellis, V. (1999). Self-similarity and multifractality of topographic surfaces at basin and subbasin scales. *Journal of Geophysical Research*, **104**(B6), 12,797–12,812.

## Appendix A

### Computation of Fractal Dimension

#### A.1 Slope-Area Plot Constructions

mfslarea.m was written to construct slope-area plots from grid DEMs. It runs in Octave/Matlab, and takes as input the output from ARC/INFO's GRIDASCII command (with header removed).

```
## SLOPEAREA.M  Simon Dadson, 16th May 2000
## Octave script to produce a slope-area plot from ARC/INFO grids of
## slope and cumulative area (named slope.<fname> and area.<fname>)
## user identifies hillslope-channel scale cutoff and
## channel network fractal dimension is computed.
## Based on algorithms in Montgomery and Foufoula-Georgiou (1993) and
## Veneziano and Iacobellis (1999)

clear;
closeplot;

disp("Slope-Area Statistics.  Simon Dadson (2000).");
disp("Reading ARC/INFO data"); fflush(stdout);

name = input("Enter a name for this analysis: ", "s");

ls1 = strcat("load area.", name);
ls2 = strcat("load slope.", name);
```

```

eval(ls1);
eval(ls2);

slope = slope ./ 100;      ## because ARC/INFO gives gradient as a percentage
area = area .* 100;      ## because GVRD DEMs have 100 sq. m. pixels
dimensions = size(area);

area_max = max(max(area));
area_min = 100;          ## i.e., grid cell area
divs = 20;              ## number of divisions
step_len = 3;          ## step length

results = zeros(divs + 1,8);
results(1:divs,2) = rot90(logspace(log10(area_min),log10(area_max),divs),3);
results(2:divs + 1, 1) = rot90(logspace(log10(area_min),log10(area_max),divs),3);
results(1,1) = 0;

results(:,3) = (results(:,1) + results(:,2)) ./ 2;

## do average for each area class
disp("Computing averaged slope-area statistics"); fflush(stdout);

for ypos = 1:step_len:dimensions(2)
    for xpos = 1:step_len:dimensions(1)
        tmp_area = area(xpos, ypos); ## to use as an index
        tmp_slope = slope(xpos, ypos);
        if tmp_area != -999900 & tmp_slope != -99.99 ## skip NODATA values
            for i = 1:divs
if tmp_area >= results(i,1) & tmp_area < results(i,2)
                results(i,4) = results(i,4) + 1;
                results(i,5) = results(i,5) + tmp_slope;
                results(i,6) = results(i,6) + tmp_slope^2;
                results(i,7) = results(i,7) + tmp_slope^3;
                results(i,8) = results(i,8) + tmp_slope^4;
            endif
        endif
    endfor
endfor
endfor

results(:,5) = results(:,5)./results(:,4);
results(:,6) = results(:,6)./results(:,4);

```

```

results(:,7) = results(:,7)./results(:,4);
results(:,8) = results(:,8)./results(:,4);

## produce slope area plot
sl_min = min(results(2:divs,8));
sl_max = max(max(results(2:divs,5:8)));

axis([1, area_max, sl_min, sl_max]);
hold on;
title(strcat("Slope-Area Plot: ", name));
xlabel("drainage area (square metres)");
ylabel("local slope m/m");
gset key left bottom title "Select Cutoff..." box;
loglog(results(2:divs,3), results(2:divs,5), "-@x;M1;");
loglog(results(2:divs,3), results(2:divs,6), "-@+;M2;");
loglog(results(2:divs,3), results(2:divs,7), "-@*;M3;");
loglog(results(2:divs,3), results(2:divs,8), "-@o;M4;");

disp("(N.B.: a message about NaN indicates that the sample size taken
from the DEM is too small for the number of class divisions chosen.
From this point on, empty class-bins will be
ignored.");fflush(stdout);

cutoff = input("Enter point for fluvial inflexion: ");
for count = 1:divs
    if cutoff >= results(count,1) & cutoff < results(count,2)
        first_point = count;
    endif
endfor

## compute fractal dimension
disp("Computing fractal dimension of river network"); fflush(stdout);

n = zeros(1,4);
sumx = zeros(1,4);
sumy = zeros(1,4);
sumxy = zeros(1,4);
sumxsq = zeros(1,4);

for m = 1:4
    for i = first_point:divs

```

```

    tmpx = log10(results(i,3));
    tmpy = log10(results(i,4+m));
    if isnan(tmpy) == 0
        sumx(m) = sumx(m) + tmpx;
        sumy(m) = sumy(m) + tmpy;
        sumxy(m) = sumxy(m) + (tmpx * tmpy);
        sumxsq(m) = sumxsq(m) + tmpx^2;
        n(m) = n(m) + 1;
    endif
endfor
endfor

xmean = zeros(1,4);
ymean = zeros(1,4);
reg_slope = zeros(1,4);
reg_intcpt = zeros(1,4);
xmean = sumx ./ n;
ymean = sumy ./ n;
reg_slope = ( (n .* sumxy) - (sumx .* sumy) ) ./ ( (n .* sumxsq) - (sumx.^2) );
reg_intcpt = ymean - (reg_slope .* xmean);
xfit = linspace(cutoff, area_max, 10);

loglog(xfit, 10.^(reg_intcpt(1) + log10(xfit) * reg_slope(1)), \
    ';Fractal Regression;');
loglog(xfit, 10.^(reg_intcpt(2) + log10(xfit) * reg_slope(2)), \
    ';;');
loglog(xfit, 10.^(reg_intcpt(3) + log10(xfit) * reg_slope(3)), \
    ';;');
loglog(xfit, 10.^(reg_intcpt(4) + log10(xfit) * reg_slope(4)), \
    ';;');

theta = -1 * reg_slope(1);
H_cn = 1 - (2 * theta)
D_cn = 2 - H_cn
tstr = ['gset key left bottom title "H_cn = ' num2str(H_cn) ' D_cn = '
    num2str(D_cn) '"];
eval(tstr);
hold off;
replot;

disp("Saving plot to ./slopearea.ps"); fflush(stdout);

```

```

gset term postscript landscape;
outstr = strcat('gset output "slopearea',name, '.ps'');
eval(outstr);
replot;
gset term x11;
disp("The next plot will be a straight line if the river network is a
self-similar fractal; if the river network is a multifractal, a
non-linear curve will be seen."); fflush(stdout);
disp("Press a key to diagnose multifractality."); fflush(stdout);
foo = kbhit();

figure(2);
axis([1,4,min(reg_slope), max(reg_slope)]);
gset xtics 1;
gset nokey;
title(strcat("Slope Area Moments: ", name));
xlabel("moment order");
ylabel("moment slope");
plot(linspace(1,4,4), reg_slope, "-@+;;");

disp("Saving plot to ./mmtslp.ps"); fflush(stdout);

gset term postscript landscape;
outstr2 = strcat('gset output "mmtslp',name, '.ps'');
eval(outstr2);
replot;
gset term x11;

disp("Done.");

```

## A.2 Construction of Hillslope Variograms

Hillslope variograms were constructed using `mfs.cc`, given below. It produces variograms of the first through fifth order moments of an elevation field defined in a file of `x, y, z` coordinates in the format output by ARC/INFO.

```

/*****/
/*mfs.cc      Simon Dadson 2000      */

```

```

/*Compute fractal dimension from */
/*ARC/INFO point coverage using */
/*variogram method */
/*****/

/* Algorithm
set up parameters from user input or command-line
read input file into linked-list of Point2d (see objgeom)
select random subsample by purging points from list
determine boundaries for log-spaced bins
compute variogram
for each point pair
compute distance (and angle)
fit pair into a bin and include it in the
    moment calculation
complete moment calculation and return
plot the variogram with octave
ask user to identify range over which to compute D (unless
    running in batch mode, in which case choose the largest range)
do linear regression & convert results to fractal statistics
write output files
*/

#include <cmath>          // (std) Basic Maths
#include <cstring>
#include <iostream.h>    // (std) i/o (suns needs .h on some includes)
#include <fstream.h>     // (std) File i/o
#include <string>        // (std) C++ string class
#include <sstream.h>     // (std) support for streaming operations on strings
#include <list>          // (std) support for linked lists
#include "objgeom"       // geometry algorithms and Point2D class
#include "stats"         // least-squares linear regression algorithms
#include "random"        // random number generator from Knuth 1981
using namespace std;

typedef list<Point2D> plist;          // Defines Point2D list type
typedef list<Point2D>::iterator plist_it; // See "objgeom" include file
typedef vector<double> dvec; // define a double-precision vector class

string runref;
int h_start, h_stop, h_num, h_tol, anisodirn, adirn_tol = 0;
float sample = -1;

```

```

void UsageMsg();
void ReadParameters();
void Summary(plist& inputdata);
void OctavePlot(dvec h, dvec gamma);
void OctaveFDPlot(dvec h, dvec gamma, dvec regstats, dvec f_bounds);
void WriteLog(const string& runref);
void WriteOut(const string& runref, const dvec& h, const dvec& gamma);
void WriteFrac(string runref, dvec lh, dvec lgamma, dvec regstats, dvec
                fracstats);
plist readinputfile(const string& runref);
dvec logspace(int start, int stop, int nsteps);
dvec variogram(const dvec hvalues, plist& inputdata);
dvec askbounds(const bool&);
dvec regress(const dvec& y, const dvec& x, const dvec& range);
dvec fconvert(dvec regstats, dvec range);
int str2int(char* s);
plist purge(plist&, float);
int fit(const double& d_ij, const dvec& hvalues);
dvec log10 (dvec v);

int main(int argc, char* argv[]) {
    bool iact = true;
    if (argc == 1) UsageMsg();
    if (string(argv[1]) == string("-i")) ReadParameters();
    else if (string(argv[1]) == ("-b")) {
        // interpret command line parameters
        if (argc != 9) UsageMsg();
        iact = false;
        runref = argv[2];
        h_start = str2int(argv[3]);
        h_stop = str2int(argv[4]);
        h_num = str2int(argv[5]);
        h_tol = str2int(argv[6]);
        anisodirn = str2int(argv[7]);
        adirn_tol = str2int(argv[8]);
    }
    else UsageMsg();
    plist biglist = readinputfile(runref);
    plist inputdata = purge(biglist, sample);
    dvec h = logspace(h_start, h_stop, h_num);
    dvec gamma = variogram(h, inputdata);

```

```

if (iact == true) OctavePlot(h, gamma);
dvec f_bounds = askbounds(iact);
dvec regstats = regress(log10(gamma), log10(h), f_bounds);
dvec fracstats = fconvert(regstats,f_bounds);
if (iact == true) OctaveFDPlot(h, gamma, regstats, f_bounds);
WriteLog(runref);
WriteOut(runref, h, gamma);
WriteFrac(runref, log10(h), log10(gamma), regstats, fracstats);
}

```

```

void ReadParameters() {
    cout << "mfs v.2 SD 21 January 2000" << endl;
    cout << "Interactive Session" << endl;
    cout << "Enter runref (input file will be <runref>.in) ";
    cin >> runref;
    cout << "Enter sample size (fraction to keep) ";
    cin >> sample;
    cout << "Enter initial value for h (h_start): ";
    cin >> h_start;
    cout << "Enter final value for h (h_stop): ";
    cin >> h_stop;
    cout << "Enter number of steps (h_num): ";
    cin >> h_num;
    cout << "Enter h-tolerance (h_tol): ";
    cin >> h_tol;
    cout << "Enter anisotropy dirn (360-degree bearing from N; -1 =
            isotropic): ";
    cin >> anisodirn;
    if (anisodirn != -1)
    {
        cout << "Enter angle tolerance (adirn_tol): ";
        cin >> adirn_tol;
    }
}

```

```

void UsageMsg() {
    cout << "mfs v2" << endl;
    cout << "Usage: fd -i (interactive mode)" << endl;
    cout << "          fd -b runref h_start h_stop h_num h_tol anisodirn
    adirn_tol (batch mode)" << endl;
    exit(0);
}

```

```

void OctavePlot(dvec h, dvec gamma){
    // Generate graphs using Octave
    string dotm = runref + ".m";
    ofstream octfile(dotm.c_str());
    octfile << "load 'mmt.'" << runref << "';" << endl;
    octfile << "h = mmt(:,1);" << endl;
    octfile << "gamma = mmt(:,2);" << endl;
    octfile << "second = mmt(:,3);" << endl;
    octfile << "third = mmt(:,4);" << endl;
    octfile << "fourth = mmt(:,5);" << endl;
    octfile << "fifth = mmt(:,6);" << endl;
    octfile << "title('Variogram: " << runref << "');" << endl;
    octfile << "xlabel('separation (h)');" << endl;
    octfile << "ylabel('moment');" << endl;
    octfile << "gset key below;" << endl;
    octfile << "loglog(h,gamma,'@33;M1;');" << endl;
    octfile << "hold on;" << endl;
    octfile << "loglog(h,second,'@33;M2;');" << endl;
    octfile << "loglog(h,third,'@33;M3;');" << endl;
    octfile << "loglog(h,fourth,'@33;M4;');" << endl;
    octfile << "loglog(h,fifth,'@33;M5;');" << endl;
    octfile << "pause" << endl;
    string callstr="octave -q " + dotm;
    system(callstr.c_str());    // calls octave to create graphs
}

void OctaveFDPlot(dvec h, dvec gamma, dvec regstats, dvec f_bounds){
    // plot fractal regression on variogram using Octave

    string fracline = runref + "fln.out";
    ofstream fln(fracline.c_str());
    for (int count = (int) f_bounds[0]; count <= (int) f_bounds[1];
        count++) {
        float x, y;
        x = h[count];
        y = pow(10, (regstats[0]*log10(x))+regstats[1]);
        fln << x << " " << y << endl;
    }
    string dotm = runref + "f.m";
    ofstream octfile(dotm.c_str());
    octfile << "load 'mmt.'" << runref << "';" << endl;
}

```

```

octfile << "load '" << runref << ".fln.out';" << endl;
octfile << "h = mmt(:,1);" << endl;
octfile << "gamma = mmt(:,2);" << endl;
octfile << "second = mmt(:,3);" << endl;
octfile << "third = mmt(:,4);" << endl;
octfile << "fourth = mmt(:,5);" << endl;
octfile << "fifth = mmt(:,6);" << endl;
octfile << "x =" << runref << ".fln(:,1);" << endl;
octfile << "y =" << runref << ".fln(:,2);" << endl;
octfile << "title('Variogram: '" << runref << "');" << endl;
octfile << "xlabel('separation (h)');" << endl;
octfile << "ylabel('moment');" << endl;
octfile << "gset key below;" << endl;
octfile << "hold on" << endl;
octfile << "loglog(h,gamma,'@33;M1;')" << endl;
octfile << "loglog(x,y,'Fractal Regression;')" << endl;
octfile << "loglog(h,second,'@33;M2;')" << endl;
octfile << "loglog(h,third,'@33;M3;')" << endl;
octfile << "loglog(h,fourth,'@33;M4;')" << endl;
octfile << "loglog(h,fifth,'@33;M5;')" << endl;
octfile << "hold off" << endl;
octfile << "pause" << endl;
// octfile << "gset term postscript landscape" << endl;
// octfile << "gset output '" << runref << ".ps'" << endl;
// octfile << "replot" << endl;
// octfile << "gset term x11" << endl;
string callstr="octave -q " + dotm;
system(callstr.c_str()); // calls octave to create graphs
}

```

```

void WriteLog(const string& runref){
// write parameters to log file
string dotlog = runref + ".log";
ofstream logfile(dotlog.c_str());
logfile << "fd.cc produced this data using the following
parameters:" << endl;
logfile << "h_start, h_stop, h_num, h_tol, anisodirn, adirn_tol,
sample" << endl;
logfile << h_start << "," << h_stop << "," << h_num << "," <<
h_tol << "," << anisodirn << "," << adirn_tol << "," <<
sample << "," << endl;
}

```

```

void WriteOut(const string& runref, const dvec& h, const dvec&
              gamma){
    // write variogram to file
    string dotout = runref + ".out";
    ofstream outfile(dotout.c_str());
    for (int count = 0; count <= (h_num-1); count++) {
        outfile << h[count] << " " << gamma[count] << endl;
    }
}

void WriteFrac(string runref, dvec lh, dvec lgamma, dvec regstats, dvec
              fracstats){
    // write fractal stats to file
    string dotfrc = runref + ".frc";
    ofstream frcfile(dotfrc.c_str());
    for (int count = 0; count <= (h_num-1); count++) {
        frcfile << lh[count] << " " << lgamma[count] << endl;
    }
    frcfile << "Regression Statistics: slope=" << regstats[0] << "
              intercept=" << regstats[1] << endl;
    frcfile << "Fractal Statistics: D=" << fracstats[0] << " Gamma(0)="
              << fracstats[1] << " Bounds " << fracstats[2] << " <= h <="
              << fracstats[3] << endl;
    cout << "Regression Statistics: slope=" << regstats[0] << " intercept="
           << regstats[1] << endl;
    cout << "Fractal Statistics: D=" << fracstats[0] << " Gamma(0)=" <<
           fracstats[1] << " Bounds " << fracstats[2] << " <= h <=" <<
           fracstats[3] << endl;
}

plist readinputfile(const string& runref){

    /*** open data file for input ***/
    string dotin = runref + ".in";
    ifstream infile(dotin.c_str());
    if (!infile)
    {
        cerr << "Cannot open: " << dotin << " for input. Aborting." << endl;
        exit(1);
    }
    else cout << "Reading Datafile: ";
}

```

```

/** create list of points and read data file into this list */
string dataline;
plist readlist;
plist_it read_it = readlist.begin();    // starts traverse of readlist on
                                        // read_it
int size = 0;                          // number of lines read
while (getline(infile, dataline))      // read one dataline
{
    size++;
    if (size == 1) /*do nothing (to eat header line) */;
    else
{
    float x, y, z, rubbish;             //temp vars to decode stream
    const char* buffer = dataline.c_str(); //dataline -> c-string for
                                        //streaming
    istringstream iss(buffer,80);
    iss >> rubbish >> z >> x >> y;      // rubbish catches ARC rec num
    Point2D temp_point(x,y,z);
    readlist.insert(read_it, temp_point);
    // cout << temp_point << endl;    // useful for testing...
}
}

/** Perform niceties and return linked list of points */
cout << size-1 << " data lines read." << endl;
return readlist;
}

dvec logspace(int start, int stop, int nsteps){
// returns a dvec of nsteps values, logarithmically spaced between
// start and stop (inclusive)
dvec logbins;
cout << "Computing logarithmically spaced h-values: ";
double lstart = (double) log10(start);
double lstop  = (double) log10(stop);
double incr   = (lstop - lstart) / ((double)nsteps - 1);
for (double lgh = lstart; lgh < lstop; lgh += incr) {
    double temp = pow(10,lgh);
    logbins.push_back(temp);
}
logbins.push_back((double)stop); //cheat for last value to cover

```

```

    // any roundoff error
    cout << "Done." << endl;
    return logbins;
}

dvec askbounds(const bool& iact){
    // ask user for bounds of fractal regression
    // choose entire range if running in command line mode
    dvec bounds;
    if (iact == true) {
        double first, last = 0;
        cout << "Enter first point for fractal estimate: ";
        cin >> first;
        cout << "Enter last point for fractal estimate: ";
        cin >> last;
        bounds.push_back(first);
        bounds.push_back(last);
    }
    else {
        bounds.push_back(0);
        bounds.push_back(h_num-1);
    }
    return bounds;
}

dvec regress(const dvec& y, const dvec& x, const dvec& range){
    // regress y on x over range. return slope, intercept in a dvec
    dvec xf;
    dvec yf;
    copy(xf, x, (int) range[0], (int) range[1]);
    copy(yf, y, (int) range[0], (int) range[1]);
    dvec xfsq = squares(xf);
    dvec yfsq = squares(yf);
    dvec xyfprod = products(xf, yf);
    int nx = xf.size();
    int ny = yf.size();
    if (nx!=ny) exit(0);
    double sumxf = sum(xf); // we'll use this a lot
    double slope = (ny*sum(xyfprod)-sumxf*sum(yf))/
        (nx*sum(xfsq)-sumxf*sumxf);
    double intercept = mean(yf) - (slope * mean(xf));
    dvec output;
}

```

```

    output.push_back(slope);
    output.push_back(intercept);
    return output;
}

dvec fconvert(dvec regstats, dvec range){
    // convert regression results into fractal statistics
    dvec fracstats;
    fracstats.push_back(3 - regstats[0]);
    fracstats.push_back(regstats[1]);
    fracstats.push_back(range[0]);
    fracstats.push_back(range[1]);
    return fracstats;
}

int str2int(char* s) {
    // return an integer representation of a numeric string
    int sign = 1;
    int value = 0;
    int place = 1;
    int length = strlen(s);
    if (length > 10) {
        cout << "str2int failed: number too large" << endl;
        exit(1);
    }
    for (int counter = 0; counter < length; counter++) {
        if (s[counter]=='-') {
            sign = -1;
            place++;
        }
        else {
            int number = int(s[counter])-48;
            if ((number >= 0) && (number <= 9)) {
                int power = (int)pow(10,(length - place));
                value += number * power;
                place++;
            }
            else {
                cout << "str2int: non-integer argument." << endl;
                exit(1);
            }
        }
    }
}

```

```

    }
    value *= sign;
    return value;
}

plist purge(plist& input, float proportion) {
    // create subsample of points based on user specified proportion
    // if running in batch mode, always choose 2000 points (if available)
    plist purged = input;
    int listsize = input.size();
    if (proportion == -1) proportion = ((float)2000 / (float)listsize);
    if (proportion > 1) proportion = 1;
    cout << "Original datafile has: " << listsize << " points." << endl;
    cout << "Random sampling..." << endl;
    Random choose;
    for (int count = 1; count <= listsize*(1-proportion); count++) {
        int purgelocn = choose.integer(purged.size()-1,0);
        plist_it del_it = purged.begin();
        for (int count2 = 1; count2 <= purgelocn; count2++) {
            ++del_it;
        }
        purged.erase(del_it);
    }
    cout << "Subsample of: " << purged.size() << " points taken." << endl;
    return purged;
}

```

```

dvec variogram(const dvec hvalues, plist& inputdata){
    // return variogram of inputdata over bins hvalues.
    // see thesis for details of this algorithm
    // (write out all moments for plotting (added 25/05/00))

    int hsize = hvalues.size();
    dvec N;
    dvec first;
    dvec second;
    dvec third;
    dvec fourth;
    dvec fifth;
    dvec gamma;
}

```

```

// fill N, sum and gamma with zeros
for (int count = 0; count < hsize; count++) {
    N.push_back(0);
    first.push_back(0);
    second.push_back(0);
    third.push_back(0);
    fourth.push_back(0);
    fifth.push_back(0);
    gamma.push_back(0);
}
// attempt to fit each point into an h-bin
for (plist_it t1=inputdata.begin(); t1 != inputdata.end(); ++t1)
{
    plist_it temp = inputdata.begin();    // allow advance by one next line
    temp = t1;
    temp ++;
    for (plist_it t2=temp; t2 != inputdata.end(); ++t2)
        { // Finds distance and angle between point pairs.
double d_ij = dist(*t1,*t2);
double theta_ij = angle(*t1,*t2);
double zdiff_ij = zdiff(*t1,*t2);
if (anisodirn != -1) {
    cout << "I have not yet learned to make anisotropic
variograms!" << endl;
    exit(1);
}
else {
    int location = fit(d_ij,hvalues);
    if (location != 9999) {
        first[location] += zdiff_ij;
        second[location] += pow(zdiff_ij, 2);
        third[location] += pow(zdiff_ij, 3);
        fourth[location] += pow(zdiff_ij, 4);
        fifth[location] += pow(zdiff_ij, 5);
        N[location]++;
    }
}
}
}

for (int count = 0; count < hsize; count++) {
    if (N[count] > 100) {

```

```

    gamma[count] = first[count]/(N[count]);
    second[count] /= N[count] * 2;
    third[count] /= N[count] * 3;
    fourth[count] /= N[count] * 4;
    fifth[count] /= N[count] * 5;
}
else {
    cout << "warning: only " << N[count] << " input points in h_bin "
<< count << "." << endl;
}
}

string outfile = "mmt." + runref;
ofstream out(outfile.c_str());
for (int count = 0; count < hsize; count++) {
    out << hvalues[count] << " " << gamma[count] << " " <<
second[count] << " " << third[count] << " " << fourth[count] << "
" << fifth[count] << endl;
}

return gamma;
}

int fit(const double& d_ij, const dvec& hvalues) {
    // return the array index in hvalues into which a point with
    // separation distance d_ij should be placed.
    int location;
    bool housed = false;
    int count = 0;
    while (housed == false) {
        double h = hvalues[count];
        if ((d_ij <= (h + h_tol)) && (d_ij >= (h - h_tol))) {
            location = count;
            housed = true;
        }
        if (count == hvalues.size()) {
            location = 9999;
            housed = true;
        }
        count++;
    }
    return location;
}

```

```

}

dvec log10 (dvec v) {
    // return log10 of a dvec (in one go)
    // same as octave/matlab
    dvec logs;
    int v_size = v.size();
    for (int i = 0; i < v_size; i++) {
        logs.push_back(log10(v[i])); // take log10 of each element
    }
    return logs;
}

/*****
/* class defn for Point2D          */
/* generic functions on plists    */
*****/

class Point2D
{ friend double dist(const Point2D&, const Point2D&);
  friend double angle(const Point2D&, const Point2D&);
  friend double zsq(const Point2D&, const Point2D&);
  friend double zdiff(const Point2D&, const Point2D&);
  friend istream& operator>>(istream& istr, Point2D& p);
  friend ostream& operator<<(ostream& ostr, const Point2D& p);
public:
    Point2D(double x, double y, double z) { _xcoord = x; _ycoord = y;
        _zvalue = z; }
    Point2D() { _xcoord = -9999; _ycoord = -9999; _zvalue = -9999; }
    double xpos() { return _xcoord; }
    double ypos() { return _ycoord; }
    double zval() { return _zvalue; }
    void print() { cout << "(" << _xcoord << ", " << _ycoord << ") = "
        << _zvalue;}

private:
    double _xcoord;
    double _ycoord;
    double _zvalue;
};

// streaming operators (to write to screen and read from keyboard)

```

```

ostream& operator<<(ostream& ostr, const Point2D& p) {
    if ((p._xcoord == -9999) || (p._ycoord == -9999)) ostr
        << "Position Undefined";
    else if (p._zvalue == -9999) ostr << "(" << p._xcoord << ","
        << p._ycoord << ") = no value";
    else ostr << "(" << p._xcoord << "," << p._ycoord << ") = " << p._zvalue;
    return ostr;
}

```

```

istream& operator>>(istream& istr, Point2D& p) {
    // format for input is x,y,z
    char bin;    // used to eat ( and ,
    istr >> p._xcoord;
    if (istr.peek()=='(',')') istr >> bin >> p._ycoord;
    if (istr.peek()=='(',')') istr >> bin >> p._zvalue;
    return istr;
}

```

```

double dist(const Point2D& p1, const Point2D& p2){
    // Finds straight line distance from p1 to p2
    if ((p1._xcoord == -9999) || (p1._ycoord == -9999) ||
        (p2._xcoord == -9999) || (p2._ycoord == -9999)) return -9999;
    else
    {
        double xdtemp = p2._xcoord - p1._xcoord;
        double ydtemp = p2._ycoord - p1._ycoord;
        xdtemp *= xdtemp;
        ydtemp *= ydtemp;
        double dtemp = sqrt(xdtemp+ydtemp);
        return dtemp;
    }
}

```

```

double angle(const Point2D& p1, const Point2D& p2){
    if ((p1._xcoord == -9999) || (p1._ycoord == -9999) ||
        (p2._xcoord == -9999) || (p2._ycoord == -9999)) return -9999;
    else
    {
        // Finds 360 degree bearing from North of pt 2 from pt 1
        double xdtemp = p2._xcoord-p1._xcoord;
        double ydtemp = p2._ycoord-p1._ycoord;
        double theta=0;
    }
}

```

```

// Correct for quadrant to get radian bearing from north
if (xdtemp > 0) theta = 1.57079632679 - atan(ydtemp/xdtemp); // = pi/2
else if (xdtemp < 0) theta = 4.71238898038 -
                        atan(ydtemp/xdtemp); // = 3 pi by 2

// Convert to degrees
theta *= 57.2957795131; // 180 / pi (rad -> deg)
return theta;
}
}

double zsq(const Point2D& p1, const Point2D& p2){
    if ((p1._xcoord == -9999) || (p1._ycoord == -9999) ||
        (p2._xcoord == -9999) || (p2._ycoord == -9999)) return -9999;
    else
    {
        // find square of distance between two points
        double zdiff = (p1._zvalue)-(p2._zvalue);
        double zsq   = zdiff*zdiff;
        return zsq;
    }
}

double zdiff(const Point2D& p1, const Point2D& p2){
    // return absolute value of z difference

    if ((p1._xcoord == -9999) || (p1._ycoord == -9999) ||
        (p2._xcoord == -9999) || (p2._ycoord == -9999)) return -9999;
    else
    {
        double zdiff = (p1._zvalue)-(p2._zvalue);
        if (zdiff < 0) zdiff *= -1;
        return zdiff;
    }
}
}

```

## Appendix B

### Derivation of Landscape Modelling Equations

In this appendix, the rules used to represent hillslope and fluvial landscape processes are derived from assumptions about their physical mechanisms.

#### B.1 Continuity Equation

Assuming that  $Z$ , the elevation at a point in the landscape, comprises the sum of landscape elements,  $z$ :

$$Z = \int_V z d^3r \quad (\text{B.1})$$

The time variation of  $Z$  is:

$$\frac{dZ}{dt} = \frac{d}{dt} \int_V z d^3r \quad (\text{B.2})$$

$$= \int_V \left\{ \frac{\partial}{\partial t} z + \frac{\partial}{\partial x_j} (v_j z) \right\} d^3r \quad (\text{B.3})$$

Where  $v_j$  is a velocity vector of the boundary of the landscape element. Suppose that:

$$\frac{dZ}{dt} = \int_V \Phi d^3r - \int_V \frac{\partial q_j}{\partial x_j} d^3r \quad (\text{B.4})$$

i.e., that changes in elevation are due to (i) "introduction" or "removal" of landscape material,  $\Phi$ , by tectonic or fluvial processes, and (ii) flux of sediment,  $q_j$ , away from the point.

Combining B3 and B4, and applying conservation of landscape volume, the local form of the balance equation becomes:

$$\frac{\partial z}{\partial t} = -\frac{\partial}{\partial x_j} \left( \frac{\partial q_j}{\partial x_j} \right) + \Phi \quad (\text{B.5})$$

## B.2 Fluvial Transport Rule

The rationalized Bagnold equation written by Martin and Church (2000) is:

$$i_b = [\omega - \omega_0]^{3/2} [D^{1/4} / d] [1 / (\rho_r^{1/2} g^{1/4})] \quad (\text{B.6})$$

where  $i_b$  = bedload (dry) mass flux per unit width,  $\omega$  = stream power per unit width,  $\omega_0$  = critical stream power,  $D$  = characteristic grain size diameter,  $d$  = channel depth,  $\rho_r$  = relative density of sediment in water, and  $g$  = acceleration due to gravity.

Assuming  $\omega = (\rho_w g Q S) / w$ ,  $\omega_0 \approx 0$ , and  $D \approx 1$ , then this expression simplifies to:

$$i_b = 0.0139 \frac{[(\rho_w Q S) / w]^{3/2}}{d} \quad (\text{B.7})$$

which is multiplied by channel width,  $w$ , to obtain the mass flux of sediment from the model cell. This is then divided by the density of sediment,  $\rho_s$ , and the area of the channel bed (length of the channel represented in the cell ( $\approx \Delta x$ , the cell dimension) multiplied by channel width) to obtain the change in river-bed elevation for the model cell,  $\Delta z_b$ , in metres per second:

$$\Delta z_b = \frac{0.0139}{d \rho_s \Delta x} \left[ \frac{\rho_w Q S}{w} \right]^{3/2} \quad (\text{B.8})$$

Sediment transporting discharge,  $Q$ , is assumed to flow for one day each year (=86,400 s), and may be approximated from upslope area using the relation obtained by Church (1997):  $Q = k_{fl} A^{0.68}$ , where  $k_{fl}$  is a parameter to represent hydroclimate. The terms  $w$  and  $d$

are correlates of discharge, so may be approximated by the hydraulic geometry relations:  $w = 3.26 Q^{0.5}$ , and  $d = 0.42 Q^{0.4}$  (Kellerhals, in Knighton, 1998) for slopes less than 0.01:

$$\Delta z_b \text{ (per year)} = 6.44 Q^{0.35} S^{1.5} \quad (\text{B.9})$$

Note that channel change has been “smeared” over the grid cell in order to account for the fact that the river is a linear feature in an areal model unit.

For slopes greater than 0.01, the hydraulic geometry relations of Day (1969) were used:  $w = 0.95 Q^{0.76}$ , and  $d \approx 0.92 Q^{0.5}$ . A correction to the slope exponent is also made, following Martin (1998), to account for configurational stability in steep channels, giving:

$$\Delta z_b \text{ (per year)} = 40.32 Q^{0.34} S^{2.22} \quad (\text{B.10})$$

## B.3 erode2d.cc

```

/*****/
/* Landscape Evolution Model */
/* Simon Dadson 25.02.00 */
/*****/

/*****/
/* PROGRAM HEADERS */
/*****/

#include <iostream.h>
#include <cmath>
#include <fstream.h>
#include <sstream.h> // (std) support for streaming operations on strings
#include <string>
#include <vector>
#include "random"
#include "landnode"
#define SIDELENGTH 30 // length of dem grid size
#define TSTOP 100000 // length of model run (years)
#define NSIDE 50 // nodes on each side of square model domain
#define NT 10000 // number of timesteps
#define HMAX 1500 // maximum height (used in generating
// artificial topography)

using namespace std;

// Support for custom data types
typedef double Matrix[NSIDE][NSIDE];
typedef vector<double> dvec;
ofstream slidefile("lscape.sli");
ofstream slidedata("lscape.sld");
ofstream sedtransp("lscape.sed");
Random billy;
Matrix fillnumber; // keeps record of sinks in flow-routing algorithm.
Matrix channelch; // keeps record of net aggradation/degradation in channels

/*****/
/* DECLARATIONS */
/*****/
```

```

// Global Constants
const int flowback[8] = {16,32,64,128,1,2,4,8}; // look-up arrays
const int xlookup[8] = {1,1,1,0,-1,-1,-1,0}; // for flow-
const int ylookup[8] = {-1,0,1,1,1,0,-1,-1}; // routing
const double rt2 = pow(2, 0.5);
const double Pi = 3.1415927;

// Global Parameters
double K, hx0, hscale; // non-linear diffusion parameters
double r; // stability parameter for solution
double crit_prob; // critical probability for dsbl slides
double alpha = 35*(Pi/180); // repose angle for talus (radian)
double k_fluvial; // proportionality const. for fluvial transport
double hftrans; // upslope area where channel initiation occurs
double Umax; // maximum uplift rate
int slide_test_int = 1; // minimum interval between dsbl slides
string inputfile; // source of initial surface
string upliftmodel; // uplift regime
double rho_rock = 2.65; // density of rock
double rho_sed = 1.60; // density of sediment
bool diagnose = false; // turns on verbose output for diagnostic work

// Functions
void title();
void getparams();
void expnonlinear(Matrix initial, Matrix final, Matrix production,
Matrix uplift, float start, float stop, float dt, string kmodel);
void renewuplift(Matrix uplift, float t, string upliftmodel, double Umax);
void dnet(Matrix landscape, Matrix slope, Matrix rivernet, Matrix
flow, double k_fluvial, float t);
void fillsinks(Matrix landscape, Matrix flow);
void flowroute(Matrix inaltgrid, Matrix slope, Matrix outdirgrid);
void accumulate(Matrix indirgrid, Matrix outareagrid);
void Bagnold(Matrix acc, Matrix slope, Matrix transport, double k_fluvial);
void fluxsep(Matrix riverchange, Matrix riverflux, Matrix flow, float t);
void dsbl(Matrix landscape, Matrix slidescape, Matrix sliderecord,
Matrix slope, Matrix flow, float t);
void export(Matrix output, bool display, int id);
void matrix_copy(Matrix old, Matrix newmatrix);
void matrix_add(Matrix first, Matrix second);
void threshold(Matrix in, Matrix out, double cutoff);
void fill(Matrix input, string file);

```

```

void writelog();
bool is_a_zero(Matrix grid);
bool is_even(int testint);
bool is_mult(double a, double b);
int drain(Matrix ingrid, int xi, int yi);
int dsxpos(int oldx, double flowdir);
int dsypos(int oldy, double flowdir);
double volume(Matrix landscape);

/*****/
/* MAIN PROGRAM */
/*****/

int main(int argc, char* argv[]) {
    title();
    getparams();
    Matrix landscape, riverchange, slope, flow, sliderecord;
    fill(landscape, inputfile);
    fill(sliderecord, "nodata");
    fill(fillnumber, "zeros");
    fill(channelch, "zeros");
    export(landscape, diagnose, 0);
    float dt = TSTOP/NT;
    r = (K * dt) / SIDELENGTH;
    cout << "r = " << r << endl;
    // cout << "Stability parameter = " << (D * dt) / (NSIDE*NSIDE) << endl;
    int id = 1;
    writelog();
    ofstream vfile("lscape.vol");
    for (float t = 0; t <= TSTOP-dt; t+=dt) {
        if (t==0) dnet(landscape, slope, riverchange, flow, k_fluvial, t);
        if (is_mult(t, slide_test_int)) {
            Matrix slidescape;
            dnet(landscape, slope, riverchange, flow, k_fluvial, -1);
        // before sliding (to refresh)
            dsbl(landscape, slidescape, sliderecord, slope, flow, t);
            matrix_copy(slidescape, landscape);
            dnet(landscape, slope, riverchange, flow, k_fluvial, t);
        // after sliding

    }
}

```

```

Matrix output;
vfile << t << " " << volume(landscape) << endl;
Matrix uplift;
renewuplift(uplift, t, upliftmodel, Umax);
exponential(landscape, output, riverchange, uplift, t, t+dt, 1,
            "constant");
if (is_mult(t, TSTOP/40)) {
    cout << "time = " << t+(TSTOP/40) << " yrs. " << "Writing
        lscape" << id << endl;
    export(output, diagnose, id);
    export(sliderrecord, diagnose, id+50);
    id++;
}
matrix_copy(output, landscape);
// feed this iteration's output back as input for next
}
// call Octave script to display animated 3D projection of landscape
export(sliderrecord, diagnose, 99);
system("mv lscape99.out lscape.slh");
export(fillnumber, diagnose, 98);
system("mv lscape98.out lscape.snk");
export(channelch, diagnose, 97);
system("mv lscape97.out lscape.cnl");
system("./vscape 2> /dev/null");
}

// TECTONICS
void renewuplift(Matrix uplift, float t, string upliftmodel, double Umax) {
    if (upliftmodel == "none") {
        // NO UPLIFT
        for (int ypos = 0; ypos < NSIDE; ypos++) {
            for (int xpos = 0; xpos < NSIDE; xpos++) {
                uplift[xpos][ypos] = 0;
            }
        }
    }
    if (upliftmodel == "const") {
        // UPLIFT CONSTANT IN SPACE AND TIME
        for (int ypos = 0; ypos < NSIDE; ypos++) {
            for (int xpos = 0; xpos < NSIDE; xpos++) {
                uplift[xpos][ypos] = Umax;
            }
        }
    }
}

```

```

    }
}
if (upliftmodel == "thrust") {
    // THRUST
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
if (xpos < (NSIDE / 2)) uplift[xpos][ypos] = Umax;
else uplift[xpos][ypos] = 0;
        }
    }
}
if (upliftmodel == "pluton") {
    // PLUTON
    double step_diff = (Umax / NSIDE) * 2;
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
if (xpos < (NSIDE / 2)) uplift[xpos][ypos] = xpos * step_diff;
else uplift[xpos][ypos] = (2 * Umax) - (xpos * step_diff);
        }
    }
}
if (upliftmodel == "rebound") {
    // ISOSTATIC REBOUND WITH (LINEAR) DECREASING RATE
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
uplift[xpos][ypos] = Umax * ((TSTOP - t) / TSTOP);
        }
    }
}
}

```

```

// FLUVIAL PROCESSES

```

```

void dnet(Matrix landscape, Matrix slope, Matrix riverchange, Matrix
flow, double k_fluvial, float t) {

```

```

    // Compute the drainage network from elevations using D8 algorithm
    // each cell in Q is the annual average discharge at that point,
    // based on the area, integrated upslope, above the point.
    // Each cell in rivernet is the amount removed by the river (i.e.,
    // negative production of sediment)

```

```

Matrix acc, riverflux;
flowroute(landscape, slope, flow);
bool tampered = false;
double before_tamper = volume(landscape);

while (is_a_zero(flow)) {
    tampered = true;
    fillsinks(landscape, flow); // fills small sinks only
    flowroute(landscape, slope, flow);
}

if (tampered) {
    // print a warning message if any closed pits have been filled
    // as a results of enforcing the flow-routing rules
    // never happens with usual erosion rules because no
    // process is capable of producing a pit.

    double after_tamper = volume(landscape);
    if ((after_tamper - before_tamper) > (after_tamper * 0.01)) {
        cout << "lscape (dnet): WARNING! elevation field modified by >
1% to route flow" << endl;
    }
}

accumulate(flow, acc);
Bagnold(acc, slope, riverflux, k_fluvial);
fluxsep(riverchange, riverflux, flow, t);
if (t != -1) matrix_add(channelch, riverchange);
if (t = TSTOP) {
    Matrix network;
    fill(network, "nodata");
    threshold(acc, network, k_fluvial/(NSIDE * NSIDE));
    export(network, diagnose, 96);
    system("mv lscape96.out lscape.net");
}
}

void fillsinks(Matrix landscape, Matrix flow) {
    /*
    for every cell, determine its lowest neighbour
    if the lowest neighbour is higher than the original cell then
    increase the height to equal the height of the lowest neighbour

```

```

    (cannot happen with usual erosion rules)
    */

    int sinkcount = 0;
    bool sinks = true;
    while (sinks) {
        sinks = false;
        for (int ypos = 2; ypos < NSIDE-2; ypos++) { // skip the edges
            // where there is always a sink
            for (int xpos = 2; xpos < NSIDE-2; xpos++) {
                double low_height = 1000000; // a very large number
                bool is_sink = true;
                for (int neighbour = 0; neighbour < 8; neighbour++) {
                    int xj = xpos + xlookup[neighbour];
                    int yj = ypos + ylookup[neighbour];
                    if (landscape[xpos][ypos] < landscape[xj][yj]) {
                        if (landscape[xj][yj] < low_height) low_height =
                            landscape[xj][yj];
                    }
                    else is_sink = false;
                }
                if (is_sink) {
                    landscape[xpos][ypos] = low_height;
                    sinks = true;
                }
            }
        }

        // For all cells with undefined flow direction
        // find the global pour point

        double global_min = 10000; // arbitrarily large value
        for (int ypos = 2; ypos < NSIDE - 2; ypos++) {
            for (int xpos = 2; xpos < NSIDE - 2; xpos++) {
                if (flow[xpos][ypos] == 0) {
                    bool has_higher = false;
                    for (int neighbour = 0; neighbour < 8; neighbour++) {
                        int xj = xpos + xlookup[neighbour];
                        int yj = ypos + ylookup[neighbour];
                        if ((landscape[xpos][ypos] != landscape[xj][yj])
                            && (landscape[xj][yj] < global_min))
                    }
                }
            }
        }
    }

```

```

        global_min = landscape[xj][yj];
    }
    }
}

// assign global_min to all cells with undefined flow direction
for (int ypos = 2; ypos < NSIDE - 2; ypos++) {
    for (int xpos = 2; xpos < NSIDE - 2; xpos++) {
        if (flow[xpos][ypos] == 0) {
            landscape[xpos][ypos] = global_min;
            fillnumber[xpos][ypos]++;
        }
    }
}

void flowroute(Matrix inaltgrid, Matrix slope, Matrix outdirgrid) {

    // kernel of D8 flow routing algorithm.
    // Input must be a depressionless DEM; if it is not, depressions
    // are returned with zero drainage direction

    // Make edges flow outward; assumes that we are well inside area of interest
    // a two-cell strip is used because the image boundary conditions create
    // spurious features at the edge, which are of no interest
    for (int xpos = 0; xpos < NSIDE; xpos++) {
        outdirgrid[xpos][0] = 128;
        outdirgrid[xpos][1] = 128;
        outdirgrid[xpos][NSIDE-1] = 8;
        outdirgrid[xpos][NSIDE-2] = 8;
    }
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        outdirgrid[0][ypos] = 32;
        outdirgrid[1][ypos] = 32;
        outdirgrid[NSIDE-1][ypos] = 2;
        outdirgrid[NSIDE-2][ypos] = 2;
    }

    // Correct corners
    outdirgrid[0][0] = 64;
    outdirgrid[1][1] = 64;
}

```

```

outdirgrid[0][NSIDE-1] = 16;
outdirgrid[1][NSIDE-1] = 16;
outdirgrid[NSIDE-1][0] = 1;
outdirgrid[NSIDE-1][1] = 1;
outdirgrid[NSIDE-1][NSIDE-1] = 4;
outdirgrid[NSIDE-2][NSIDE-2] = 4;

// for each interior node compare its elevation with those of its neighbours
double diag_div = pow(2 * pow(SIDELENGTH, 2), 0.5); // horizontal
// length for points on diagonal
double diag_corr = SIDELENGTH / diag_div;
// correction for diagonals
for (int ypos = 2; ypos < NSIDE-2; ypos++) {
    for (int xpos = 2; xpos < NSIDE-2; xpos++) {
        double cellheight = inaltgrid[xpos][ypos];
        double local_max = 0;
        int local_flow = 0;
        for (int neighbour = 0; neighbour < 8; neighbour++) {
int xj = xpos + xlookup[neighbour];
int yj = ypos + ylookup[neighbour];
double n_slope = (cellheight - inaltgrid[xj][yj]) / SIDELENGTH;
if (is_even(neighbour)) {
    // if on a diagonal, divide by sqrt(2)
    n_slope = n_slope * diag_corr ; // correct for diagonals
}
if (n_slope > local_max) {
    local_flow = (int) pow(2, neighbour);
    local_max = n_slope;
}
else if (n_slope == local_max) {
    local_flow += (int) pow(2, neighbour);
}
}
if (local_max == 0) local_flow = 0;
slope[xpos][ypos] = local_max;
outdirgrid[xpos][ypos] = (double) local_flow;
}
} // END OF FIRST PASS

// Choose direction randomly for cells that could have several
for (int ypos = 2; ypos < NSIDE-2; ypos++) {
    for (int xpos = 2; xpos < NSIDE-2; xpos++) {

```

```

    int local_flow = (int) outdirgrid[xpos][ypos];
    if (local_flow == 0) ; // do nothing: wait for next sweep
    // assign central cell if a whole side is equally lower
    else if (local_flow == 7) local_flow = 2;
    else if (local_flow == 28) local_flow = 8;
    else if (local_flow == 112) local_flow = 32;
    else if (local_flow == 193) local_flow = 128;
    else {
// unpack flow directions and make a random choice if
// several are equally valid
dvec choices;
int pips = local_flow;
if (pips >= 128) {
    choices.push_back(128);
    pips -= 128;
}
if (pips >= 64) {
    choices.push_back(64);
    pips -= 64;
}
if (pips >= 32) {
    choices.push_back(32);
    pips -= 32;
}
if (pips >= 16) {
    choices.push_back(16);
    pips -= 16;
}
if (pips >= 8) {
    choices.push_back(8);
    pips -= 8;
}
if (pips >= 4) {
    choices.push_back(4);
    pips -= 4;
}
if (pips >= 2) {
    choices.push_back(2);
    pips -= 2;
}
if (pips >= 1) {
    choices.push_back(1);

```

```

    pips -= 1;
}

if (choices.size() == 1) local_flow = (int) choices[0];
else {
    int index = billy.integer(choices.size()-1, 0);
    local_flow = (int) choices[index];
}
    }
    outdirgrid[xpos][ypos] = local_flow;
}
} // END OF SECOND PASS

// Resolve flow for flat nodes by iteration
// flow may be to any node that has an assigned direction provided
// that node does not flow back to the original.
// the while loop may get stuck if there are extensive areas of
// perfect flatness if it does, tail_chaser bails it out and dnet
// will call fill_sinks. Few landscapes are perfectly flat
// so this is not usually a problem.

int tail_chaser = 0;
while ((is_a_zero(outdirgrid)) && (tail_chaser < NSIDE*NSIDE)) {
    tail_chaser++;
    for (int ypos = 2; ypos < NSIDE-2; ypos++) {
        for (int xpos = 2; xpos < NSIDE-2; xpos++) {
if (outdirgrid[xpos][ypos] == 0) {
            int flat_flow = 0;
            dvec flat_choices;
            for (int neighbour = 0; neighbour < 8; neighbour++) {
                int xj = xpos + xlookup[neighbour];
                int yj = ypos + ylookup[neighbour];
                if ((inaltgrid[xpos][ypos] - inaltgrid[xj][yj] == 0)
                    && (outdirgrid[xj][yj] != 0)
&& (outdirgrid[xj][yj] != flowback[neighbour])) {
                    double dirn = pow(2, neighbour);
                    flat_choices.push_back(dirn);
                }
            }
            if (flat_choices.size() == 0) flat_flow = 0; // no change
            else if (flat_choices.size() == 1) flat_flow =
                (int) flat_choices[0];

```

```

else {
    int index = billy.integer(flat_choices.size()-1, 0);
    flat_flow = (int) flat_choices[index];
}
outdirgrid[xpos][ypos] = flat_flow;
}
    }
} // END OF THIRD PASS
}
}

int drain(Matrix ingrid, int xi, int yi) {
    // find upslope area of point xi, yi on the
    // flowdirection grid ingrid
    // (recursive)
    int cellcount = 0;
    for (int neighbour = 0; neighbour < 8; neighbour++) {
        int xj = xi + xlookup[neighbour];
        int yj = yi + ylookup[neighbour];
        if ((xj > 0) && (xj < NSIDE) && (yj > 0) && (yj < NSIDE)) {
            if (ingrid[xj][yj] == flowback[neighbour]) {
                int upstream = drain(ingrid, xj, yj);
                cellcount += 1 + upstream;
            }
        }
    }
    return cellcount;
}

void accumulate(Matrix indirgrid, Matrix outareagrid) {
    // given a grid of flow-directions, return upslope area
    // of each cell
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
            outareagrid[xpos][ypos] = drain(indirgrid, xpos, ypos);
        }
    }
}

void Bagnold(Matrix acc, Matrix slope, Matrix riverflux, double k_fluvial) {

    // convert upslope area into (bedload) sediment flux

```

```

// define parameters that give regional context
double k_fl = k_fluvial;
double c_chan_maint = hftrans; // m^2 -> km^2

for (int ypos = 0; ypos < NSIDE; ypos++) {
    for (int xpos = 0; xpos < NSIDE; xpos++) {

        double upslparea = acc[xpos][ypos] * (SIDELENGTH * SIDELENGTH);
            // -> m^2
        double q = k_fl * pow(upslparea / 1000000, 0.68);

        if (upslparea >= c_chan_maint) {
if (slope[xpos][ypos] < 0.01) {
    // compute sediment transport rate using
    // normal Bagnold relation
        // (Martin and Church, 1999; Knighton, 1998)
        double pointdiff = 6.44 * pow(q, 0.35) *
            pow(slope[xpos][ypos], 1.5);
        riverflux[xpos][ypos] = pointdiff;
}

else if (slope[xpos][ypos] < 0.15) {
    // compute sediment transport rate using
    // headward Bagnold relation (Martin, 1998; Day, 1969)
        double pointdiff = 40.32 * pow(q, 0.34) *
            pow(slope[xpos][ypos], 2.22);
        riverflux[xpos][ypos] = pointdiff;
}

else {
    riverflux[xpos][ypos] = 0;
}

        }

        else riverflux[xpos][ypos] = 0;
            // no fluvial processes above channel head
        }
    }
}

void fluxsep(Matrix riverchange, Matrix riverflux, Matrix flow, float t) {

```

```

// use predicted total flux at each point and input flux from upstream
// neighbours to compute amount of erosion or deposition at each point
// change = input - output

for (int ypos = 0; ypos < NSIDE; ypos++) {
  for (int xpos = 0; xpos < NSIDE; xpos++) {

    double inflow = 0;

    for (int neighbour = 0; neighbour < 8; neighbour++) {
int xj = xpos + xlookup[neighbour];
int yj = ypos + ylookup[neighbour];
if ((xj > 0) && (xj < NSIDE) && (yj > 0) && (yj < NSIDE)) {
  if (flow[xj][yj] == flowback[neighbour]) {
    inflow += riverflux[xj][yj];
  }
}
}

    double netchange = inflow - riverflux[xpos][ypos];
    riverchange[xpos][ypos] = netchange / (SIDELENGTH * SIDELENGTH);

  }
}

// write total bedload sediment flux to sedtransp, as a row of
// NSIDE*NSIDE+1 columns
// (first column is time)
// ... records one row every time dnet is run from main.. one row
// each 100 years
if (t != -1) {
  sedtransp << t << " ";
  for (int ypos = 0; ypos < NSIDE; ypos++) {
    for (int xpos = 0; xpos < NSIDE; xpos++) {
sedtransp << riverflux[xpos][ypos] << " ";
    }
  }
  sedtransp << endl;
}
}

```

```

// DIFFUSION

void expnonlinear(Matrix initial, Matrix final, Matrix production,
  Matrix uplift, float start, float stop, float dt, string kmodel) {

  // Explicit 2-D solver for non-linear landscape diffusion equation
  // with initial surface, from start to stop with interval dt
  // using kmodel specified and including terms for
  // production of material (tectonics, rivers etc.) (spatially dependent)
  // with results written to final (which must exist when fn is called)

  Matrix U, V;
  matrix_copy(initial,U);
  fill(V, "zeros");
  int toggle = 1;
  double K1 = K / 2;
  for (float time = start; time <= stop; time+=dt) {
    if (toggle == 1) {

      // set boundary using method of images
      for (int ypos = 1; ypos < NSIDE-1; ypos++) {
// left and right bnds
V[0][ypos] = U[2][ypos];
V[NSIDE-1][ypos] = U[NSIDE-3][ypos];
      }
      for (int xpos = 1; xpos < NSIDE-1; xpos++) {
// top and bottom bnds
V[xpos][0] = U[xpos][2];
V[xpos][NSIDE-1] = U[xpos][NSIDE-3];
      }
      // set diagonal image at corners
V[0][0] = U[2][2];
V[0][NSIDE-1] = U[2][NSIDE-3];
V[NSIDE-1][0] = U[NSIDE-3][2];
V[NSIDE-1][NSIDE-1] = U[NSIDE-3][NSIDE-3];

      for (int ypos = 1; ypos < NSIDE-1; ypos++) {
for (int xpos = 1; xpos < NSIDE - 1; xpos++) {

// Compute some space derivatives

```

```

double XA = (U[xpos+1][ypos] - U[xpos][ypos]) / SIDELENGTH;
double YA =
    (U[xpos+1][ypos+1] + U[xpos][ypos+1] -
     U[xpos+1][ypos] - U[xpos][ypos])
    / (2*SIDELENGTH);

double XB = (U[xpos][ypos] - U[xpos-1][ypos]) / SIDELENGTH;
double YB =
    (U[xpos-1][ypos+1] + U[xpos][ypos+1] -
     U[xpos-1][ypos] - U[xpos][ypos])
    / (2*SIDELENGTH);

double YC = (U[xpos][ypos+1] - U[xpos][ypos]) / SIDELENGTH;
double XC =
    (U[xpos][ypos] + U[xpos][ypos+1] -
     U[xpos-1][ypos+1] - U[xpos-1][ypos])
    / (2*SIDELENGTH);

double YD = (U[xpos][ypos] - U[xpos][ypos-1]) / SIDELENGTH;
double XD =
    (U[xpos][ypos-1] + U[xpos][ypos] -
     U[xpos-1][ypos-1] - U[xpos-1][ypos])
    / (2*SIDELENGTH);

// compute sediment flux using non-linear function

double slpA = XA + YA;
double slpB = XB + YB;
double slpC = XC + YC;
double slpD = XD + YD;

double QA = K1 * (1 + tanh( (abs( slpA ) - hx0) / hscale)) * (slpA);
double QB = K1 * (1 + tanh( (abs( slpB ) - hx0) / hscale)) * (slpB);
double QC = K1 * (1 + tanh( (abs( slpC ) - hx0) / hscale)) * (slpC);
double QD = K1 * (1 + tanh( (abs( slpD ) - hx0) / hscale)) * (slpD);

// compute elevation change

double diffuse = (dt/SIDELENGTH)*(QA-QB+QC-QD);

if ((slpA > 1.6) || (slpB > 1.6) ||

```

```

        (slpC > 1.6) || (slpD > 1.6)) {
    // turn off diffusion on slopes > ~ 60 degrees
    diffuse = 0;
}

double change = diffuse + (production[xpos][ypos] * dt) +
    (uplift[xpos][ypos] * dt);
V[xpos][ypos] = U[xpos][ypos] + change;
}
    }
}

if (toggle == -1) {

    // set boundary using method of images
    for (int ypos = 1; ypos < NSIDE-1; ypos++) {
// left and right bnds
U[0][ypos] = V[2][ypos];
U[NSIDE-1][ypos] = V[NSIDE-3][ypos];
    }
    for (int xpos = 1; xpos < NSIDE-1; xpos++) {
// top and bottom bnds
U[xpos][0] = V[xpos][2];
U[xpos][NSIDE-1] = V[xpos][NSIDE-3];
    }
    // set diagonal image at corners
U[0][0] = V[2][2];
U[0][NSIDE-1] = V[2][NSIDE-3];
U[NSIDE-1][0] = V[NSIDE-3][2];
U[NSIDE-1][NSIDE-1] = V[NSIDE-3][NSIDE-3];

    for (int ypos = 1; ypos < NSIDE-1; ypos++) {
for (int xpos = 1; xpos < NSIDE - 1; xpos++) {
// Compute some space derivatives

double XA = (V[xpos+1][ypos] - V[xpos][ypos]) / SIDELENGTH;
double YA =
    (V[xpos+1][ypos+1] + V[xpos][ypos+1] -
     V[xpos+1][ypos] - V[xpos][ypos])
    /(2*SIDELENGTH);

double XB = (V[xpos][ypos] - V[xpos-1][ypos]) / SIDELENGTH;

```

```

double YB =
    (V[xpos-1][ypos+1] + V[xpos][ypos+1] -
     V[xpos-1][ypos] - V[xpos][ypos])
    /(2*SIDELENGTH);

double YC = (V[xpos][ypos+1] - V[xpos][ypos]) / SIDELENGTH;
double XC =
    (V[xpos][ypos] + V[xpos][ypos+1] -
     V[xpos-1][ypos+1] - V[xpos-1][ypos])
    /(2*SIDELENGTH);

double YD = (V[xpos][ypos] - V[xpos][ypos-1]) / SIDELENGTH;
double XD =
    (V[xpos][ypos-1] + V[xpos][ypos] -
     V[xpos-1][ypos-1] - V[xpos-1][ypos])
    /(2*SIDELENGTH);

// compute sediment flux using non-linear function

double slpA = XA + YA;
double slpB = XB + YB;
double slpC = XC + YC;
double slpD = XD + YD;

double QA = K1 * (1 + tanh( (abs( slpA ) - hx0) / hscale)) * (slpA);
double QB = K1 * (1 + tanh( (abs( slpB ) - hx0) / hscale)) * (slpB);
double QC = K1 * (1 + tanh( (abs( slpC ) - hx0) / hscale)) * (slpC);
double QD = K1 * (1 + tanh( (abs( slpD ) - hx0) / hscale)) * (slpD);

// compute elevation change

double diffuse = (dt/SIDELENGTH)*(QA-QB+QC-QD);

if ((slpA > 1.6) || (slpB > 1.6) ||
    (slpC > 1.6) || (slpD > 1.6)) {
    // turn off diffusion on slopes > ~ 60 degrees
    diffuse = 0;
}

double change = diffuse + (production[xpos][ypos] * dt) +
    (uplift[xpos][ypos] * dt);

```

```

    U[xpos][ypos] = V[xpos][ypos] + change;
}
    }
}
    toggle *= -1;
}
if (toggle == 1) matrix_copy(U, final);
else if (toggle == -1) matrix_copy(V, final);
else {
    cout << "lscap (expsolv): internal error" << endl;
    exit(1);
}
}
}

```

```
// DEEP-SEATED BEDROCK LANDSLIDES
```

```
void dsbl(Matrix landscape, Matrix slidescape, Matrix sliderecord,
Matrix slope, Matrix flow, float t) {
```

```

    // modify landscape to create slidescape based on the occurrence of
    // deep-seated bedrock landslides; slides occur at random.
    // most likely sites are identified using model nodes where gradient
    // elevation and length of slope are highest

```

```

    // put time in slidefile
    slidefile << t << endl;

```

```

    // Generate sliding probabilities
    Matrix slideprob;
    matrix_copy(landscape, slideprob);
    matrix_copy(landscape, slidescape);
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
            slideprob[xpos][ypos] *= slope[xpos][ypos];
        }
    }
}

```

```

    // Make a list of probabilities
    Landnode nodelist[NSIDE*NSIDE];
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {

```

```

        nodelist[xpos*NSIDE+ypos].x(xpos);
        nodelist[xpos*NSIDE+ypos].y(ypos);
        if ((xpos <= 5) || (xpos >= NSIDE-5)
            || (ypos <= 5) || (ypos >= NSIDE-5)) {
// zero probability of sliding near edge
nodelist[xpos*NSIDE+ypos].elev(0);
        }
        else {
nodelist[xpos*NSIDE+ypos].elev(slideprob[xpos][ypos]);
        }
    }
}

// Sort the list of probabilities
sort(nodelist, (NSIDE*NSIDE));

// Choose those probabilities greater than a critical threshold
int limit = -1;
bool steep_end = false;
for (int probcount = 0; probcount < (NSIDE*NSIDE); probcount++) {
    if (nodelist[probcount] <= crit_prob) limit++;
}
// cout << limit << "," << (NSIDE*NSIDE)-1 << endl;
if (limit == (NSIDE*NSIDE)-1) {
    // bail out here if we arent having a slide
    slidefile << "no slide occurred" << endl;
}

else {

    //otherwise choose a slide location at random
    int xpos, ypos, cougar;
    bool selected = false;
    cougar = billy.integer((NSIDE*NSIDE)-1, limit);
    xpos = (int) nodelist[cougar].xcoord();
    ypos = (int) nodelist[cougar].ycoord();
    slidefile << nodelist[cougar] << endl;
    slidedata << nodelist[cougar] << " ";
    sliderecord[xpos][ypos] = 1;

    // Calculate the maximum failure size at the chosen site
    double current_height = slidescape[xpos][ypos];

```

```

slidefile << "current height = " << current_height << endl;
int nextx = dsxpos(xpos, flow[xpos][ypos]);
int nexty = dsypos(ypos, flow[xpos][ypos]);
slidefile << "Next cell: " << nextx << ", " << nexty << endl;
double next_cell_height = slidescape[nextx][nexty];
slidefile << "Next cell height = " << next_cell_height << endl;
double biggest_slide =
    current_height - next_cell_height - (SIDELENGTH * tan(alpha));
if (biggest_slide < 0) biggest_slide = 0; // if slope not steep enough
slidefile << "biggest slide = " << biggest_slide << endl;

double ndmag = 1; // all available relief fails
slidefile << "weibull proportion = " << ndmag << endl;
double slide_size = ndmag*biggest_slide;
slidefile << "actual slide size = " << slide_size << endl;

// Begin volume check during slide
double pre_vol = volume(landscape);

// reduce elevation at failure site
slidescape[xpos][ypos] -= slide_size;

// decide how much talus we have generated
double stash = slide_size*(rho_rock / rho_sed); // allow for density change
slidefile << "available regolith = " << stash << endl;
double ttl_dep_vol = stash; // track ttl amount moved
double ds_difference = SIDELENGTH*tan(alpha); // difference to maintain
// angle of repose

// move the material downslope until it runs out
int dscout = 0; // how many times have we moved downslope with debris
int current_xpos = nextx; // start at node downslope of originating node
int current_ypos = nexty;

while (stash > 0) {
    slidefile << "Moved downstream" << endl;
    slidefile << "at location: " << current_xpos << ", "
        << current_ypos << " ";
    dscout++;
    double current_height = slidescape[current_xpos][current_ypos];
    int new_xpos = dsxpos(current_xpos, flow[current_xpos][current_ypos]);
    int new_ypos = dsypos(current_ypos, flow[current_xpos][current_ypos]);

```

```

        slidefile << "done backflow" << endl;
        if ((new_xpos <= 0) || (new_xpos >= NSIDE-1) ||
            (new_ypos <= 0) || (new_ypos >= NSIDE-1)) {
// in case we go off the edge
slidefile << "lscap (dsbl): a freak slide has occurred: " << stash
        << " units of sediment have left the model domain.";
stash = 0;
        }

        else {
double ds_height = slidescape[new_xpos][new_ypos];
double ideal_height = ds_height + ds_difference;
double required_height = ideal_height - current_height;
slidefile << "downslope height = " << ds_height << " ;ideal_height = "
        << ideal_height << " ; reqdheight = " << required_height
                << endl;

if (required_height >= 0) {
// if we are in a depositional node

slidefile << "Depositing: ";
if (stash >= required_height) {
// if there is enough material deposit to ideal_height
slidescape[current_xpos][current_ypos] = ideal_height;
stash -= required_height;
slidefile << required_height << " units" << endl;
}

else {
// otherwise just leave the entire amount that we have
slidescape[current_xpos][current_ypos] += stash;
slidefile << stash << " units" << endl;
stash = 0;
}
}

else {
// otherwise the slide scours the downslope area
slidefile << "Scouring ";
slidescape[current_xpos][current_ypos] = ideal_height;
double surplus = slidescape[current_xpos][current_ypos] -
                ideal_height;

```

```

slidefile << surplus << " units" << endl;
surplus *= (rho_rock / rho_sed);
stash += surplus;
ttl_dep_vol += surplus; // add to total slide volume
}
current_xpos = new_xpos;
current_ypos = new_ypos;
    }

}

double post_vol = volume(slidescape); // to check conservation of volume
slidefile << "Total Volume: " << ttl_dep_vol << endl;
slidefile << "Runout length: " << dscount << endl;
slidefile << ttl_dep_vol << " " << dscount << endl;
slidefile << "Pre-vol: " << pre_vol << " . Post-vol: " << post_vol << endl;
slidedata << ttl_dep_vol << " " << dscount << endl;
cout << "DSBL @ (" << xpos << "," << ypos << "); Vol = " <<
ttl_dep_vol * (SIDELENGTH * SIDELENGTH) << " m^3 ; Runout = " <<
dscount * SIDELENGTH << " m." << endl;
}
}

```

```
// INPUT-OUTPUT FUNCTIONS
```

```

void export(Matrix output, bool display, int id) {

// write a file containing output, named lscape<id>.out.
// if the flag display is true, the output is also written to the screen.

string prefix = "lscape";
int tens = (int)(id/10);
int units = id - (10*tens);
char tnum = tens + 48;
char unum = units + 48;
string filename = prefix + tnum + unum + ".out";
ofstream outfile(filename.c_str());
for (int ypos = 0; ypos < NSIDE; ypos++) {
    for (int xpos = 0; xpos < NSIDE; xpos++) {
        outfile << output[xpos][ypos] << " ";
    }
}
}

```

```

    }
    outfile << endl;
}
if (display == true) {
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
            cout << output[xpos][ypos] << " ";
        }
        cout << endl;
    }
}
}

void fill(Matrix input, string file) {

    // Fill a matrix, input, with values taken from the file
    // if file is "random" or "zeros" then no file is read
    // and the matrix is filled with uniform random values or
    // zeros accordingly

    if (file == "random") {
        for (int ypos = 0; ypos < NSIDE; ypos++) {
            for (int xpos = 0; xpos < NSIDE; xpos++) {
                input[xpos][ypos] = billy.integer(HMAX,0);
            }
        }
    }
    else if (file == "zeros") {
        for (int ypos = 0; ypos < NSIDE; ypos++) {
            for (int xpos = 0; xpos < NSIDE; xpos++) {
                input[xpos][ypos] = 0;
            }
        }
    }
    else if (file == "nodata") {
        for (int ypos = 0; ypos < NSIDE; ypos++) {
            for (int xpos = 0; xpos < NSIDE; xpos++) {
                input[xpos][ypos] = -9999;
            }
        }
    }
    else if (file == "sine") {

```

```

    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
if ((xpos == 0) || (ypos == 0) || (xpos == NSIDE - 1) ||
    (ypos == NSIDE - 1)) input[xpos][ypos] = 0;
else input[xpos][ypos] =
    (2*(NSIDE)) - ((xpos) * sin((4*Pi*ypos)/(NSIDE)));
        }
    }
}
else if (file == "comb") {
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
if ((xpos == 0) || (ypos == 0) || (xpos == NSIDE - 1) ||
    (ypos == NSIDE - 1)) input[xpos][ypos] = 0;
else {
    if (is_even(xpos)) input[xpos][ypos] = (xpos*NSIDE + ypos);
    else input[xpos][NSIDE - ypos - 1] = (xpos*NSIDE + ypos);
}
        }
    }
}
else if (file == "ramp") {
    double step_diff = HMAX / (NSIDE / 2);
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
if (xpos < (NSIDE / 2)) input[xpos][ypos] = xpos * step_diff;
else input[xpos][ypos] = (2 * HMAX) - (xpos * step_diff);
        }
    }
}
else if (file == "test") {
    // test data from Jenson and Domingue (1988)
    double testinput[NSIDE][NSIDE]
    =      {{ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    { 0,778,765,750,740,747,759,765,766,769,776,786,795, 0},
    { 0,770,758,745,737,741,751,753,761,777,789,802,814, 0},
    { 0,777,763,747,736,735,743,750,767,787,806,820,832, 0},
    { 0,786,767,750,737,733,739,752,769,785,797,808,822, 0},
    { 0,794,773,756,741,733,733,744,759,772,779,789,806, 0},
    { 0,799,782,763,750,737,733,733,745,757,767,782,801, 0},
    { 0,802,788,771,761,751,736,733,738,751,764,779,798, 0},
    { 0,799,790,780,772,762,746,733,737,754,770,784,794, 0},

```

```

    { 0,811,799,787,771,757,741,728,730,745,765,779,783, 0},
    { 0,823,807,790,774,762,748,733,725,733,750,764,763, 0},
    { 0,830,814,801,787,776,761,743,728,725,737,748,751, 0},
    { 0,822,818,811,801,791,776,757,739,726,725,735,751, 0},
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}};
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
// rotate 90 degrees because it easier to enter like that above!
input[xpos][ypos] = testinput[ypos][xpos]; // / 832;
        }
    }
    else {
        ifstream infile(file.c_str());
        if (!infile)
        {
cerr << "lscape (fill): cannot open input file: " << file << endl;
exit(1);
        }
        cout << "lscape (fill): reading initial values from: " << file << endl;
        string dataline;
        for (int ypos = 0; ypos < NSIDE; ypos++) {
            {
getline(infile, dataline);
double temp;
const char* buffer = dataline.c_str(); //dataline -> c-string for streaming
istream iss(buffer, 10000);
for (int xpos = 0; xpos < NSIDE; xpos++) {
    iss >> temp;
    input[xpos][ypos] = temp;
}
            }
        }
    }
}

```

```
// AUXILLIARY FUNCTIONS
```

```

double volume(Matrix landscape) {
    // compute the approximate volume of landscape
    double landvolume = 0;

```

```

double base_area = SIDELENGTH * SIDELENGTH;
for (int ypos = 0; ypos < NSIDE - 2; ypos+=2) {
    for (int xpos = 0; xpos < NSIDE - 2; xpos+=2) {
        double ttl_elev = landscape[xpos][ypos] + landscape[xpos+1][ypos] +
landscape[xpos][ypos+1] + landscape[xpos+1][ypos+1];
        landvolume += (0.25*ttl_elev*base_area);
    }
}
return (landvolume);
}

void title() {
    cout << "erode2d: non-linear diffusion with threshold landsliding" << endl;
    cout << "Diffusive-advective landform evolution model" << endl;
    cout << "Written by Simon Dadson, April 2000" << endl;
}

void writelog() {
    // write log file
    ofstream logfile("lscape.log");
    logfile << "ERODE2D Log File lscape.log: " << endl;
    logfile << "Initial surface = " << inputfile << endl;
    logfile << "SIDELENGTH = " << SIDELENGTH << endl;
    logfile << "TSTOP = " << TSTOP << endl;
    logfile << "NSIDE = " << NSIDE << endl;
    logfile << "NT = " << NT << endl;
    logfile << "HMAX = " << HMAX << endl;
    logfile << "D = " << D << endl;
    logfile << "K, hx0, hscale" << endl;
    logfile << K << " " << hx0 << " " << hscale << endl;
    logfile << "stability parameter = " << r << endl;
    logfile << "Pcrit = " << crit_prob << endl;
    logfile << "alpha = " << alpha << endl;
    logfile << "K_fluvial = " << k_fluvial << endl;
    logfile << "hftrans = " << hftrans << endl;
    logfile << "uplift model = " << upliftmodel << endl;
    logfile << "max uplift = " << Umax << endl;
    logfile << "slide test interval = " << slide_test_int << endl;
    logfile << "rho_rock = " << rho_rock << endl;
    logfile << "rho_sed = " << rho_sed << endl;
    logfile << "diagnose = " << diagnose << endl;
}

```

```

int dsxpos(int oldx, double flowdir) {
    // find x coordinate of point downstream of oldx, given flowdir
    int newx = -1;
    for (int nc = 0; nc < 9; nc++) {
        if (flowdir == pow(2, nc)) {
            newx = oldx + xlookup[nc];
            // back to front !
        }
    }
    if (newx == -1) {
        cout << "lscapex (dsxpos): error routing sediment" << endl;
        exit(1);
    }
    return (newx);
}

```

```

int dsypos(int oldy, double flowdir) {
    // find y coordinate of point downstream of oldy, given flowdir
    int newy = -1;
    for (int nc = 0; nc < 9; nc++) {
        if (flowdir == pow(2, nc)) {
            newy = oldy + ylookup[nc];
        }
    }
    if (newy == -1) {
        cout << "lscapex (dsypos): error routing sediment" << endl;
        exit(1);
    }
    return (newy);
}

```

```

void matrix_copy(Matrix old, Matrix newmatrix) {
    // copy old into new
    // new must exist prior to the call

    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
            newmatrix[xpos][ypos]=old[xpos][ypos];
        }
    }
}

```

```

    }
}

void matrix_add(Matrix first, Matrix second) {
    // add second to first, result is in first,
    // second is unchanged
    // both must exist prior to the call

    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
            first[xpos][ypos]+=second[xpos][ypos];
        }
    }
}

bool is_even(int testint) {
    // return true if testint is even; false if odd
    float teststat = 0;
    teststat = testint%2;
    if (teststat != 0) return false;
    else if (teststat == 0) return true;
    else exit(1);
}

bool is_mult(double a, double b) {
    // return true if a is a multiple of b
    double teststat = 0;
    teststat = int(a)%int(b);
    if (teststat != 0) return false;
    else if (teststat == 0) return true;
    else exit(1);
}

void getparams() {
    cout << "Enter name of input file: " ;
    cin >> inputfile;
    int location;
    cout << "Use nonlinear diffusion rule for GVRD (1) or QCI (2): ";
    cin >> location;
    if (location == 1) {
        K = 0.12;
        hx0 = 0.81;
    }
}

```

```

    hscale = 0.32;
}
else if (location == 2) {
    K = 0.43;
    hx0 = 1.06;
    hscale = 0.19;
}
else {
    cout << "Invalid location" << endl;
    exit(1);
}

cout << "Enter uplift style (none, const, thrust, pluton, rebound): ";
cin >> upliftmodel;
cout << "Enter maximum uplift rate (metres per year): ";
cin >> Umax;
cout << "Enter coefficient for fluvial transport (k): ";
cin >> k_fluvial;
cout << "Enter hillslope-fluvial transition
        (in upslope area units (m2)) ";
cin >> hftrans;
cout << "Enter minimum dsbl recurrence interval (years): ";
cin >> slide_test_int;
cout << "Enter critical slide probability (angle * elevation): ";
cin >> crit_prob;
}

bool is_a_zero(Matrix grid) {
    // return true if grid contains any zero, false otherwise.
    bool found_one = false;
    for (int ypos = 0; ypos < NSIDE; ypos++) {
        for (int xpos = 0; xpos < NSIDE; xpos++) {
            if (grid[xpos][ypos] == 0) found_one = true;
        }
    }
    return found_one;
}

void threshold(Matrix in, Matrix out, double cutoff) {
    // produce out from in, replacing every value greater than cutoff with 1
    // and every value lower or equal to with -9999 (arc NODATA)
    for (int ypos = 0; ypos < NSIDE; ypos++) {

```

```
for (int xpos = 0; xpos < NSIDE; xpos++) {  
    if (in[xpos][ypos] > cutoff) out[xpos][ypos] = 1;  
    else out[xpos][ypos] = -9999;  
}  
}  
}
```