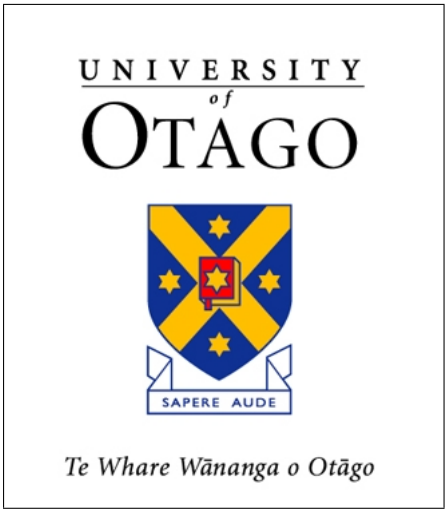
Regular Hierarchical Surface Models

A conceptual model of scale variation in a GIS and its application to hydrological geomorphometry

Joseph W. Wright

August 2017



A thesis submitted for the degree of Doctor of Philosophy
at the University of Otago, Dunedin, New Zealand.

The precision of naming takes away from the uniqueness of seeing.- Pierre Bonnard

In as far as a digital representation of space is a description of reality, that description is incomplete if it does not include its scale or resolution.- Michael F. Goodchild

Abstract

Environmental and geographical process models inevitably involve parameters that vary spatially. One example is hydrological modelling, where parameters derived from the shape of the ground such as flow direction and flow accumulation are used to describe the spatial complexity of drainage networks. One way of handling such parameters is by using a Digital Elevation Model (DEM), such modelling is the basis of the science of geomorphometry.

A frequently ignored but inescapable challenge when modellers work with DEMs is the effect of scale and geometry on the model outputs. Many parameters vary with scale as much as they vary with position. Modelling variability with scale is necessary to simplify and generalise surfaces, and desirable to accurately reconcile model components that are measured at different scales. This thesis develops a surface model that is optimised to represent scale in environmental models.

A Regular Hierarchical Surface Model (RHSM) is developed that employs a regular tessellation of space and scale that forms a self-similar regular hierarchy, and incorporates Level Of Detail (LOD) ideas from computer graphics. Following convention from systems science, the proposed model is described in its conceptual, mathematical, and computational forms. The RHSM is assessed using adaptions of the geomorphometric algorithms flow direction and flow accumulation.

The RHSM has an implicit data model that utilises a variation of Middleton and Sivaswamy (2001)'s intrinsically hierarchical Hexagonal Image Processing referencing system, which is here generalised for rectangular and triangular geometries. The RHSM provides a simple framework to form a pyramid of coarser values in a process characterised as a scaling function. In addition, variable density realisations of the hierarchical representation can be generated by defining an error value and decision rule to select the coarsest appropriate scale for a given region to satisfy the modeller's intentions.

The effects of scale and geometry on the anistropy and accuracy of model results are analysed on dispersive and concentrative cones, and Light Detection And Ranging (LiDAR) derived surfaces of the urban area of Dunedin, New Zealand. The RHSM modelling process revealed aspects of the algorithms not obvious within a single geometry, such as,

the influence of node geometry on flow direction results, and a conceptual weakness of flow accumulation algorithms on dispersive surfaces that causes asymmetrical results. In addition, comparison of algorithm behaviour between geometries undermined the hypothesis that variance of cell cross section with direction is important for conversion of cell accumulations to point values.

The RHSM development was informed by a categorisation of Geographical Information Science (GISc) surfaces within a practical framework of geometry, structure, interpolation, and data model. The positioning of the RHSM within this broader framework made it easier to adapt algorithms designed for other surface models to conform to the new model. The ability to analyse algorithms for scale and geometry and adapt algorithms within a unified conceptual framework offers deeper insight into algorithm behaviour than previously achieved. The deconstruction of algorithms into geometry neutral forms and the application of scaling functions are important contributions to the understanding of spatial parameters within GISc.

Acknowledgments

Thank you, thank you, thank you. Much admiration is deservedly given for the substantial efforts of so many people to build the rich ecosphere of open source software for scientific study. A special mention for the contributors to Python, PyTables and NumPy which together perform the heavy lifting for the analysis in this thesis.

Thank you for the guidance and knowledge of the examiners whose insight and encouragement helped me considerably to complete this thesis.

And I forever remain appreciative of my supervisors' capacity to ignore the whooshing sound of my deadlines flying by.

Contents

List of Tables

List of Figures

Nomenclature

2.5-D Surface	A functional surface model in a 3-D reference frame (x, y, z), where there is a maximum of one z value for each (x, y) location.
3-D Surface	A surface modelled as the two dimensional faces of solid three dimensional objects.
Aggregation number (α)	The number of levels of an RHSM dataset that are stored in the array component of the array tree.
Anistropy	The antonym of isotropy. A directionall non-uniform model is anistropic.
Aperture ($\mathfrak{a}$)	The ratio of area of a parent tile over the area of a child tile in a hierarchical tesselation.
Base level	In an RHSM dataset the base level is the finest Level of Detail in a pyramid of resolutions.
Cardinality (n)	The number of tiles in a tessellation or nodes in a network. When applied to RHSM, cardinality refers to the number of cells in the finest Level of Detail.
Cell size	In a square raster, cell size is the dimension of a single cell in x and y, assuming that the grid is parallel with the axes. More generally, in regular tesselations the cell size is the distance between edge adjacent neighbours.
Compression	In the context of the analysis in this thesis, compression is the difference of cardinality, expressed as a percentage, between the base resolution and a sparse realisation of an RHSM dataset.
Conforming	When describing networks, a conforming network is one in which all edges separate a maximum of two faces and there are no nodes part way along any edge.

Connectivity — The number of edges meeting at a node in a network graph, or tiles touching at a vetex in a tessellation.

Digital Elevation Model (DEM) — Digital representation of an elevation surface. Can be raster, TIN, contour or other surface types. DTMs and DSM are subsets of DEMs.

Digital Surface Model (DSM) — A subset of DEM that includes objects like trees and buildings in the surface.

Digital Terrain Model (DTM) — A subset of DEM that excludes objects like trees and buildings from the surface.

Distributed model — A model of a process where at least one variable in the model varies in at least two geographic dimensions.

Geographic Information Science (GISc) — The practise of GIS in accordance with scientific method.

Geographic Information System (GIS) — A database and user interface that stores and manages information associated with positions; and facilitates analysis, display, and sharing of spatial data. Also used informally to describe the associated discipline and practise of using a GIS.

Hierarchical data format version-5 (hdf5) — A hierarchical file structure that has been optimised for fast retrieval of data from large datasets. Implemented in Python by the PyTables module.

Hydrological and Hydraulic (H&H) — Hydrological and Hydraulic models combine surface runoff generation (Hydrological) with the physical properties of water (Hydraulic) to model catchment behaviour.

Isotropy — A measure of the uniformity of a spatial dataset or model with direction. A directionally uniform model is isotropic.

Lattice — A surface model in which data points are separate entities.

Level (λ) — The number of Levels of Detail in an RHSM. Also the number of digits in an RHSM ordinate.

Level Of Detail (LOD) — A distributed model that contains multiple scales and a mechanism to select appropriate scale.

Level of Detail (LoD)	A specific scale within a Level Of Detail model.
Network graph	A surface model in which data points are connected by edges.
Pyramid	A series of representations of a given region at successively coarser resolutions.
Scale	A combination of resolution, extent, and zoom on a continuum from coarse to fine for which an environmental model is expected to operate.
Sparse realisation	A variable density tessellation produced from an RHSM dataset.
Specific Catchment Area (a)	The limit as contour length approaches 0 of the area of catchment contributing flow to a length of contour line.
Tessellation	A collection of tiles that collectively fills space without gaps or overlaps.
Valence	See connectivity.

1. Introduction

This chapter introduces the thesis in five sections: Section §1.1 describes the topic, Section §1.2 portrays the need for the research, Section §1.3 specifies the research's objectives and questions, Section §1.4 summarises the thesis and its contributions, and Section §1.5 outlines the structure of the document.

1.1. Topic

The role that Geographical Information System (GIS) data structures perform in environmental modelling is developing. This development is driven by rapidly improving digital information about the world and rising expectations about what Geographic Information Science (GISc) can achieve to inform and assist communities' understanding of their place in the world. The advent of airborne and terrestrial LiDAR, and photogrammetric technologies; the collation of fine scale spatial information in Geographical Information System (GIS) databases, and exponentially increasing computing power means that it is now possible to make effective use of these data to represent complex urban and natural environments at the appropriate scale to resolve intricate spatial processes.

For instance, the management of stormwater is currently undergoing a step change; it is moving rapidly away from relatively simplistic approaches that only estimate the peak volumetric surface water flow rate expected from a particular storm event, and moving towards an approach that is fully dynamic and capable of predicting the time varying quantity and quality of stormwater at many points in the network over longer time frames that may contain several storm events (Kampf and Burges, 2007). Such an approach demands that our diverse environments be fully represented as spatially-rich 3D surfaces in numerical models so that the effects on the generation and transport of stormwater of all elements of our built and natural environment can be fully appreciated.

Therefore, GIS surface models are required that can represent the complexity required to accurately represent the environment across numerous scales from very coarse to very fine, whilst retaining sufficient simplicity of structure to reduce the demands of environmental

modelling to a practical computational workload. The following pages develop, describe, and assess a surface model that exploits an implicit hierarchical structure to support variability of key parameters with scale.

1.2. Need for Research

To sustain higher density communities, the human race adapts the natural environment to its needs by altering environmental systems. Urban consent authorities, planners, surveyors, engineers, and other professions are concerned with mitigating the effect of urbanisation on natural systems such that urban development proceeds in a sustainable fashion. A clear understanding of how urban settlements work, including their interaction with the surrounding natural and built environments, is fundamental, in order to ensure that existing and future urban development continues to be viable for future generations. Also needed are engineering tools for producing effective, appropriate and resilient solutions to the demands of urbanisation.

Continuous distributed two dimensional (2-D) numerical models are useful tools to understand, develop and manage changing environments because they increase the capacity of models to capture complex interactions through the inclusion of detailed geospatial information. One example is Hydrological and Hydraulic (H&H) models, which utilise detailed surface models to estimate surface water flow depth and discharge at any point in the landscape.

Reported global economic losses from flooding have increased in recent decades from an annual median average of about $0.5 billion in the 1980s to around $20 billion in the first decade of this century (Lamond and Penning-Rowsell, 2014). These rising costs are driven in part by urbanisation, which concentrates population and wealth in cities. The United Nations (2010) estimates that more than 50% of the world's population currently live in urban areas and projects this figure to grow to nearly 70% by 2050. The rising economic cost from flooding is likely to accelerate as global urban populations increase and the effects of climate change on the amount of stormwater that 'runs off' the surface of the built environment during rainstorm events become more pronounced.

The academic discipline of quantitative analysis of the hydrological behaviour of topographic surfaces is known as geomorphometric hydrological analysis (O'Callaghan and Mark, 1984; Gallant and Hutchinson, 2011). Geomorphometric hydrological analysis has evolved from the foundational work of Bevan and Kirkby (1979), Peucker and Douglas (1975), and others to become packages of modelling tools, such as ArcHydro (Maidment, 2002), TAS (Lindsay, 2005), and TauDEM (Tarboton, 2008), which are used to analyse

hydrological outcomes on basin-wide spatial scales. Most of the progress in distributed hydrological modelling until recently has been restricted to relatively coarse scale (10m - 100m spatial resolution) modelling of undeveloped and agricultural catchments. Wilson et al. (2000) states that DEMs with a spatial resolution of 2m – 10m are required for important hydrological processes in agricultural landscapes. Urban areas typically contain significant drainage features that occur at even finer spatial scales.

Traditional geographic data representations were geared toward the representation of static situations on a planar surface at a specific scale (Wilson et al., 2000). Modelling hydrologic behaviour using a regular single resolution Digital Elevation Model (DEM) requires a spatial resolution sufficient to capture the smallest significant features. However, computer resources may not be used efficiently if the spatial resolution required to resolve fine scale surface features is applied uniformly across the entire study area. In addition, methods for determining the appropriate scale to use in H&H models are not clearly defined nor universally agreed upon. Capturing processes that operate at multiple spatial scales in environmental models entails a trade-off between the complexity required to incorporate fine and coarse scale features, and the need for the computational efficiency. Multi-scale surface models offer a way to resolve this conflict.

Recent developments from the field of 2-D flood inundation modelling have sought to address the issues of scale in H&H models by applying new approaches to surface modelling, examples include triangular unstructured meshes (Tsubaki and Fujita, 2010), adaptive regular grids (Wang and Liang, 2011), and multi-level coarse grids (Chen et al., 2012). This research extends and advances these developments by utilising techniques from Level of Detail (LOD) models originally developed for computer graphics (Danovaro et al., 2006) that generate realisations of surfaces using hierarchical data structures. These techniques have not previously been applied to drainage models and have the potential to determine the appropriate scale to capture hydrological processes in complex environments.

1.3. Objectives and Research Questions

The long-term objective of this research is to contribute to the development of simulation tools and decision support systems for urban hydrology that planners, engineers and decision makers can use for efficient and effective water resource management. The research described in this thesis was motivated by a perceived need to improve the accuracy of geomorphometric catchment analysis of urban environments by the development and utilisation of innovative digital surface models. The central objectives of this research were;

1. to identify the characteristics of a surface model that considers scale and geometry in geomorphometric catchment analysis,
2. to describe a surface model that meets these criteria and develop a mathematical and computational framework to integrate the proposed surface into a GIS, and
3. to evaluate the performance of the new surface for geomorphometric catchment analysis with respect to scale and geometry.

The above objectives were refined in the research questions below.

1. What are the characteristics of GIS surface models and how can they be catalogued and understood?
2. What kind of surface model provides the ability to model hydrological processes that occur at different scales in different areas?
3. How can a surface model that models processes at different scales in different areas be implemented?
4. Can the algorithms of flow direction and accumulation be generalised to hexagonal, triangular, and variable resolution surfaces?
5. How does the surface approximation in geomorphometric catchment analysis affect model outputs?
6. Can we resolve complex overland drainage networks using multi-scale surface modelling?

1.4. Summary of contribution

The task of modelling scale variation is an important and challenging task for GISc (Goodchild, 2011). This research approaches the challenge of scale by combining diverse concepts from academic literature, primarily, geomorphometric hydrological analysis (O'Callaghan and Mark, 1984; Gallant and Hutchinson, 2011), hierarchical referencing systems (Middleton and Sivaswamy, 2001) and Level of Detail (LOD) modelling (de Floriani et al., 2005). These fields are combined within a classical systems modelling framework (Barnsley, 2007).

An initial task that underpinned this research was a broad review of academic literature covering three increasingly specific topics, which are presented in Part I of this thesis. Most general of the topics covered was the science and art of utilising spatial data for environmental models, a more specific topic was the surface models used in computer

models, and most specifically was the algorithms associated with geomorphometric hydrological analysis. These topics were related to each other by the topic of scale and the complexity of modelling variables that vary with scale. This question of scale was an important factor for the development of the Regular Hierarchical Surface Model (RHSM), which is described in Part II of this thesis.

The surface model described here adapts its spatial resolution to the significance of the underlying processes whilst retaining a computationally simple and uniform data structure. The proposed surface is evaluated for the task of hydrological geomorphometry: the analysis of surface shape to predict flow direction and accumulation. Classical techniques to determine these parameters were deconstructed and generalised to support hexagonal as well as rectangular sampling, and multi-scale environments. The decomposition of surface modelling and geomorphometric algorithms into geometry and scale independent forms provided interesting insight into the behaviour of these representations of physical processes and has great potential to guide the refinement of surface based analysis in GISc.

The techniques introduced in this thesis are promising tools for addressing the challenge of modelling scale variation. This thesis is an important step in the development of simulation tools for urban hydrology because it investigates promising techniques of surface modelling that had not previously been evaluated for drainage models.

1.5. Outline

In addition to this introduction and a brief concluding chapter, the thesis is divided into three parts and each part is divided into three chapters, giving a total of 11 chapters. The structure and narrative of the thesis is given below.

Part I, Background, reviews current knowledge in academic literature, and organises the diverse ideas and sources.

Chapter 2, Distributed environmental modelling, discusses modelling of spatial data. An approach is described to form computer models in phases: conceptual, mathematical, and computational. The concept of distributed hydrological models is discussed, scale in a GIS is defined, and the challenges of modelling distributed variables that vary with scale are identified.

Chapter 3, Digital Elevation Models, sets out a structured approach to catalogue surface models by considering separately data sources, structures, geometries, interpolation methods, and data models. Multi-scale models are also discussed. The systematic approach to describing surfaces is useful to understand algorithms for deriving surface parameters.

Chapter 4, Hydrological geomorphometry, describes geomorphometric catchment analysis including several different approaches for determining flow accumulation and flow direction, hydrological conditioning, and also references some other important objects and variables from geomorphometry.

Part II, The Regular Hierarchical Surface Model, is the core of original work in this thesis. It presents the conceptual, mathematical and computational model of the proposed Regular Hierarchical Surface Model (RHSM).

Chapter 5, Conceptual Regular Hierarchical Surface Model, proposes a surface model that supports scale variability. The motivation is to build hydrological models that consider variability of scale. The surface model has regular structure, regular subdivision of scale and space, and efficiently supports multi resolution pyramids and graph structured variable resolution representations. Due to the regular structure and inherently hierarchical referencing system, both position and hierarchy are implicit. The structure supports the regular tessellations of space: rectangular, hexagonal, and triangular tiles.

Chapter 6, Mathematical Regular Hierarchical Surface Model, describes the indexing methodology and the mathematical structure of the RHSM.

Chapter 7, Computational Regular Hierarchical Surface Model, describes the file structure, GIS integration, and catalogues the tools developed to interact with the RHSM.

Part III, Hydrological applications, applies the proposed surface model to geomorphometric hydrological analysis and evaluates its performance.

Chapter 8, Hydrological modelling in the RHSM, describes how the variables flow direction and flow accumulation were adapted to the RHSM and how flow direction is used to control the modelling resolution within the multi-resolution structure. New algorithms are developed to support flow direction and accumulation determination on variable resolution surfaces, triangular surfaces, and hexagonal surfaces. These techniques are developed from the $D8$ and $D\infty$ methods for square grid rasters.

Chapter 9, Evaluation, examines the efficacy of routing on variable resolution surfaces by comparison with full resolution outputs on a number of surfaces, both mathematical and LiDAR generated representations of urban surfaces. The new routing methods are tested on mathematical surfaces where the desired results can be determined analytically: convergent cone and divergent cone. Consistency between scales is tested and hexagonal, rectangular, and triangular tessellations are compared.

Chapter 10, Discussion, reviews the RHSM and the results of the evaluation in the context of relevant academic literature. Future work is proposed, including improving efficiency, refining flow accumulation algorithms; applying surface models to deterministic, distrib-

uted unsteady flow models; exploring other algorithms, and developing more flexible discretisations of scale and space.

Chapter 11, Conclusion, summarises and concludes the thesis.

Part I.

Background

This part consists of three chapters that serve as a background for the subsequent chapters of this thesis by reviewing and summarising selected literature from the fields of distributed environmental modelling (Chapter 2), GIS Digital Elevation Models (Chapter 3), and geomorphometric analysis (Chapter 4). The themes of this literature review are developed into a conceptual and mathematical model for a proposed new surface data structure in Part II.

2. Distributed environmental modelling

This chapter is an overview of distributed environmental modelling with an emphasis on surface hydrology and scale effects.

2.1. Model development

This section introduces the purpose of modelling and the stages of model development, and then identifies the types of models used for hydrological modelling.

2.1.1. Purpose of modelling

Producing data for decision makers that is authoritative and understandable is of vital importance for resource management and environmental decision making in general, and in water resource management in particular (Liu et al., 2008). Environmental models can be used to enhance understanding, run simulations, and make predictions about environmental systems (Barnsley, 2007). Hydrological models are used as technical information to support decision making but, due to their technical complexity, are not easily included in participatory decision support. However, to be informative to a collection of actor groups involved in decision making it is insufficient that model outputs are only understandable to technically proficient individuals.

To be relevant, model outputs need to address the needs of an audience. They need to provide information pertinent to their intentions, this may involve making predictions in situations where no measurements exist, giving insight into mechanisms or communicating knowledge of how systems work. However, all models are approximations of reality and as such do not accurately represent every aspect of a system. Assumptions are required in environmental models to reflect the limits of knowledge about the system (Barnsley, 2007). Therefore, outputs can never be forecast with certainty (Chow et al., 1988). Models are

approximate tools that continue to develop and modify in response to knowledge gained from observing the natural system being modelled (Oreskes and Belitz, 2001).

2.1.2. Stages of modelling

To assist the development of mathematical environmental models, Barnsley (2007) emphasizes three stages of development, the conceptual model, the mathematical model, and the computational model. The conceptual model is represented verbally, the mathematical model through equations, and the computational model is the implementation of the mathematical model within a software package or programming language. To these three stages of model development, some authors add the perceptual model or mental model that informs the development of conceptual models. When assessing the outputs of environmental models, people will compare the results of the model to their mental model of the world. Where there is a discrepancy, the assumptions of the model may be questioned or the model may inform a change in that person's mental model (Meadows et al., 2004).

The mediums in which each stage of the modelling process are represented have inherent limitations. Conceptual models are constrained by the symbolic language and diagrammatic systems they are communicated in. Mathematical models follow the structures of mathematics, computational models are limited by both the syntax of the programming language they are written in and the digital hardware on which the software is run. It is reasonable to ask why one should restrain the model with the formalisms of conceptual and mathematical models on its path from mental model insight to computational model code. A persuasive response to this question is that in a well developed model, the assumptions made in forming the conceptual model are stated clearly and the mathematical model is rigorously defined.

Computational models could be developed directly from the conceptual model without stating the mathematical model, however, this approach would make it difficult to assess the fidelity of the computational model. In addition, without a mathematical model, it would be impossible to assess and correct for numerical errors in the computational model (Clark and Kavetski, 2010). The conceptual model is critical to the overall effectiveness of the model and its transportability. The field of systems science describes methods that facilitate the development of conceptual models. Systems science has been criticised for misuse in social sciences but has proven effective in describing elements of natural systems and interactions between nature and society (Anderberg, 2005).

2.1.3. Types of models

Hydrological processes can be represented with physical scale models or abstract mathematical models (Chow et al., 1988). Barnsley (2007); Skidmore (2002) and others define several distinctions between different types of mathematical environmental models. Mathematical models can be developed empirically or from theory (Clarke, 1973). Empirically derived models, which are based on experimentally determined relationships, may not be as easily generalized to other study areas compared to models developed from theory. Hydrological models derived from equations describing conservation of mass, momentum, and/or energy are called physically based models (Kampf and Burges, 2007). If it is sufficient that a model reproduces the behaviour of a process without understanding the underlying causal structure itself, an empirically determined model is sufficient. Beven (2001) states that one of the arguments in favour of distributed hydrological models is that they may be more "realistic" than simpler models that are calibrated to historical data in a curve-fitting exercise, with no guarantee that they will do as well in simulating responses in other periods or other conditions.

Mathematical models may be discrete or continuous. Discrete models are solved using difference equations and divide the process into discrete steps. Continuous models are solved using differential equations and model processes instantaneously. Mathematical models can be solved numerically or analytically, that is they can be solved by an iterative process or through a precise mathematical formulation. Mathematical models may be linear or non-linear (Clarke, 1973). Mathematical models are linear if some derivative of their governing equations are constant. Table 2.1 groups the choice between these classes of mathematical model by the modeller's preference for principle or practicality.

Table 2.1.: Classes of mathematical model organised by emphasis for principle or practicality. The columns are not exclusive. A specific mathematical model may cross between columns, include elements of both or be on a continuum between the row dichotomies.

	Principled	Practical
Development	Theoretical	Empirical
Equation Form	Continuous	Discrete
Equation Shape	Linear	Non-linear
Solution	Analytical	Numerical

When considering hydrological models Chow et al. (1988) identifies three choices that a modeller makes when choosing the approach to take to modelling hydrological systems. Chow et al. (1988) observe that hydrological models are functions of randomness, space and time. Consequently, modellers can choose which of these parameter classes vary within the model.

Randomness Models may be deterministic or stochastic, stochastic models provide probabilities of certain outcomes. Stochastic models are appropriate to represent processes that appear random due to a lack of knowledge of system inputs and processes, or because of genuine randomness within the system.

Space Models may be divided into lumped models, where model parameters are held constant over large areas, or distributed models where parameters are allowed to vary on a finer scale, thereby modelling the spatial variation of inputs more closely. Distributed models can vary in one, two, or three dimensions of geographical space. Distributed hydrological models are sometimes known as variable contributing area models. Variable contributing area models acknowledge that some parts of the terrain will contribute more flow both above and below ground because they have a propensity to saturate (Quinn et al., 1995; Beven and Kirkby, 1979). Distributed hydrological models are discussed in more detail in Section §2.2.

Time Models may be constant in time or may vary with time. For instance, a time variant run-off model would acknowledge that the portion of precipitation contributing to surface flow increases as a precipitation event continues and storage mechanisms become saturated (Chow et al., 1988). Models that are invariant and models that vary with time are sometimes called unsteady and steady respectively. Hydrological models may be used to represent a single storm event or be continuous over an extended period of time (Booth, 1991). Extended period models can effectively capture the effect of past precipitation on storage mechanisms.

Combined, these three choices form a classification tree for hydrological models. A diagram of this tree is reproduced from Chow et al. (1988) as Figure 2.1. Building a model that addresses all three of these sources of variation is difficult. Therefore, most models consider only one or two of the three sources of variation. In Section §5.1 a new branch will be introduced to the classification tree by considering a hydrological model that includes variability as a function of scale.

2.2. Distributed hydrological models

A specific type of model called a distributed hydrological model is the subject of this section. The term "distributed" is defined as it relates to modelling and several approaches to distributed hydrological modelling are described. An aligned discipline to the field of hydrological modelling is hydraulic modelling, which is concerned primarily with the flow and conveyance of fluids, including water and sewerage. Models that include both

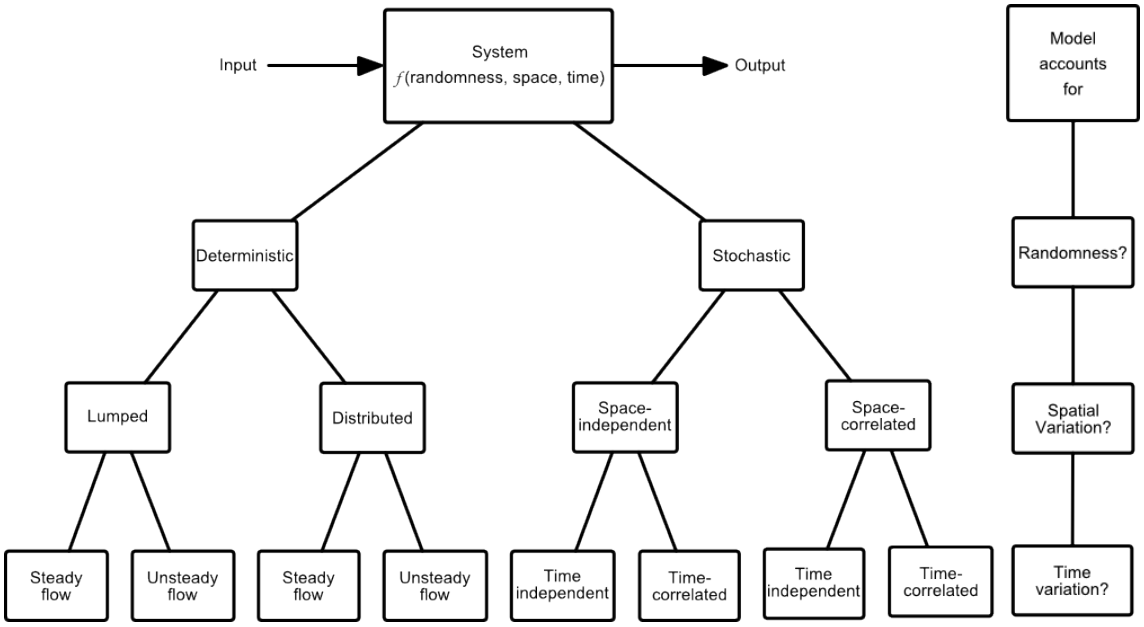


Figure 2.1.: Classification of hydrological models reproduced from Chow et al. (1988).

hydrology and hydraulic concepts are common and can be described as Hydraulic and Hydrological models (H&H).

2.2.1. Definition of distributed model

As defined by Chow et al. (1988), distributed models are those that vary in space. However, the usage of the term distributed hydrological model varies in both academic and professional literature. For instance, Chow et al. (1988) use the term very generally to refer to any models that support spatial variability in one or more model parameters. Kampf and Burges (2007) emphasise the common usage of the term to refer to any model that represents the spatial variability and pathways of water through a catchment. Therefore, the emphasis is on the representation of pathways in two or three dimensions in the model, including 1-D paths in two dimensions. Singh and Woolhiser (2002), on the other hand, apply a more limited definition of a distributed hydrological model that requires that all aspects of the model support spatial variability. Todini (1988) divides distributed approaches into "distributed integral," which is actually a network of connected lumped models and distributed differential models which include distributed flow calculations.

It is more accurate to think of distributed models as being on a continuum from, but not including, fully lumped models in which all variables are averaged to a single value with no spatial variation to fully distributed models in which all variables vary in three dimensions of space. Although the exact meaning of the term and whether a particular model is or is not defined as distributed is of little relevance when considering the efficacy of that model

for its intended purpose, for clarity in this thesis the term distributed hydrological model refers to models that represent hydrological processes where at least one variable in the model varies in at least two geographic dimensions(x, y). It is worth noting that not just the model inputs and calculations but also model outputs may or may not be distributed.

The subsurface and surface may be considered separate domains within a distributed hydrological model. However, there is flow between the two domains. Surface flows infiltrate the subsurface and subsurface flows re-emerge to contribute to surface flows. Models that account for this two-way interaction are described as coupled. A coupled model may have different dimensionality in different sub domains. For instance, the subsurface may be 3-D, the surface hill flow 2-D, and surface channel flow 1-D.

Distributed hydrological models are a branch of Computational Fluid Dynamics and are increasingly used for scientific research, hydrological forecasting, and engineering design applications. Distributed models of surface hydrology in particular have great potential for applications such as non-point source pollutant transport, hydrological responses to land use or land cover changes, land-atmosphere interactions, erosion, and sediment transport (Kampf and Burges, 2007). Routing of surface water is one of the more promising applications of distributed hydrological modelling, particularly in areas where gravity effects dominate inertial effects because in such areas surface hydrology is driven by elevation, which can be measured fairly accurately over extensive areas. Modellers can use the spatially explicit structure of distributed models to incorporate spatial data from Geographic Information Systems (GIS), remote sensing, and geophysical techniques. For example, distributed models can incorporate topographic features, the effects of shade and aspect on hydrological response, geologic and land cover variability, surface friction, and depression storage, which all have spatial variability. Distributed models are currently used to incorporate spatially explicit radar rainfall data, snow cover extent, soil moisture, and land surface temperature data into hydrological models (Kampf and Burges, 2007).

2.2.2. Numerical approaches to distributed hydrological modelling

Mathematical hydrological models typically rely on the definition of a control volume: a reference frame in three dimensions through which a fluid flows. Frequently, physically based distributed hydrological models are based on Reynold's transport theorem. Freeze and Harlan (1969) and Freeze (1974) introduced and developed a blueprint for physically based surface-subsurface distributed models using partial differential equations (PDEs) of fluid flow in three spatial dimensions and time. Solving the governing equations for representations of water flow in a distributed model analytically is only possible in a few prismatic geometries with homogeneous isotropic domains and steady state flow,

or by making simplifying assumptions to derive closed form solutions of the governing conservation equations (Kampf and Burges, 2007).

Boundary value problems generally employ discretisation methods to solve numerical approximations of continuous functions. The most common discretisation methods associated with solving PDEs are the finite difference method, finite volume method, and finite element method. Finite difference methods represent space in 1, 2, or 3 dimension domains by a series of points in a structured mesh (i.e. a rectangular point set, see Subsection 3.5.2). In this method, the values of functions are represented at each grid point.

Finite volume methods are also discrete, but they divide the domain into volumes and calculate fluxes averaged across the surface of the volumes, which can be part of an unstructured mesh (e.g. a TIN, see Subsection 3.4.2). It is unnecessary to know the path of the fluid through the control volume, it is sufficient to model the properties of the fluid at the boundary of the control volume (Chow et al., 1988). The governing equation, expressed in the partial differential equations, is reformulated, at each computational cell, into a set of linear algebraic equations (Andersson et al., 2011). Finite element methods can also accommodate unstructured meshes with complex geometries, but in contrast to the discrete finite difference and finite volume methods, they use continuous base functions to locally describe the solution of the governing equation to be approximated, with the simplest of these functions representing a plane (Kampf and Burges, 2007). The finite-element method aims to minimize the difference between the exact solution and the collection of base functions (Andersson et al., 2011).

2.2.3. The de St. Venant equations

The physically based equations typically used to represent surface water flow are known as the de St. Venant (1871) equations for shallow water flow. The de St. Venant equations include equations of continuity and momentum and have 1-D and 2-D forms. The de St. Venant equations are derived by depth integrating the Navier Stokes equations, which define single phase fluid flow.[1] The de St. Venant equations assume that if a volume of fluid is shallow relative to the wavelength of the wave phenomenon, the horizontal velocity

[1] Stelling and Zijlema (2003) take a different approach to providing solutions to the Navier Stokes equations. Rather than depth integrating the Navier Stokes equations, the Stelling method solves 3-D Reynolds-averaged Navier-Stokes equations. Therefore, the Stelling method is suitable for the entire range of water depths relative to wavelength and for simulating wave disturbance in fluid with rotation and shear. Stelling and Zijlema (2003) demonstrate that their method can accurately simulate relatively short waves using a small number of vertical layers (in the order of 1–3). The grid based 2-D component of the commercial hydrological modelling package TUFLOW is based on the Stelling method (BMT WBM, 2015).

field is nearly constant throughout the depth profile and the velocity in the z direction is small. The numerical solution to the de St. Venant equations will determine the velocity of flow from cell to cell, if the elevation of the ground is changing the vertical velocity can then be calculated using the continuity equation.

The full de St Venant equations in 1 or 2-D are known as the dynamic wave scheme. The dynamic wave scheme can capture back water effects such as a rise in the surface elevation of flowing water upstream as a result of an obstruction to flow downstream. The 1-D dynamic wave is used in several modelling programs including HEC-RAS (Tate et al., 2002) and WASH 123D (Yeh et al., 1998). Dynamic Wave scheme equations are difficult to solve, particularly in 2-D, so are frequently simplified using the diffusion and kinematic wave approximations.

The diffusion wave approximation assumes that the inertial terms of the 2-D momentum equations are negligible and can be ignored. If configured similarly to the governing equations for subsurface flow, the diffusion wave scheme facilitates coupled surface-subsurface flow computation (Panday and Huyakorn, 2004). The diffusion wave approximation does not model back water effects in time and is inaccurate for fast rising hydrographs particularly on flat riverbeds (Fread, 1993). The diffusion wave scheme is effective in most hill slope surface flow situations. Ponce (1978) gives criteria for assessing whether a diffusion wave model for open channel flow is appropriate.

Kinematic wave approximations only consider the effects of gravity and friction on flow. In this approach, the friction force and gravity force balance each other, therefore, the hydraulic gradient is assumed to be equal to the topographic slope. Due to the reduced number of unknowns, kinematic approximations are computationally robust (Kampf and Burges, 2007). Kinematic approaches do not account for backwater effects and are not well suited to represent flows at low slopes or in areas with high lateral inflows (Freeze, 1974). Kinematic approximations are appropriate only for steep terrain, where water flow directions are largely governed by topography (Kampf and Burges, 2007). Morris and Woolhiser (1980) provide guidelines to indicate when the kinematic approximation is valid.

2.2.4. Alternative methods of distributed hydrological modelling

Numerical solutions to de St Venant PDEs require significant computer resources and may be impractical for fine meshes over large domains. The mathematical techniques used to solve the PDEs are typically implemented by specialists and applied by modellers who may not understand the simplifications that have been applied. An alternative approach

is to simplify overland flow direction calculations by creating flow networks based on topography using elevation data. Overland flow can be routed through such a network in one dimension, with a specified width or area accounting for the second dimension (Kampf and Burges, 2007). This technique falls under the category of geomorphometry, is discussed in detail in Section §4.1, and forms the basis of the hydrological model developed in Chapter 8.

Alternatively, empirical schemes could be used. Empirical approaches are based on experimentally determined relationships such as linear regressions (Kampf and Burges, 2007). Some examples of empirical approaches to hydrological modelling are reservoir schemes, base flow recession curves, and the Soil Conservation Service curve number method, which estimates infiltration losses to the subsurface by empirically derived standard curves for different soil types and moisture conditions (Williams and LaSeur, 1976).

2.3. Scale and aggregation

Scale is an essential dimension of spatial data that transcends specific applications (Atkinson and Tate, 2000). Scale is so critical to representations of space that contain spatial variability that Goodchild (2001) observed that "In as far as a digital representation of space is a description of reality, that description is incomplete if it does not include its scale or resolution." Scale is a complex concept that requires careful definition to avoid confusion.

This section defines scale in any spatial or temporal dimension as it used in this thesis, and introduces the concept of appropriate scale. Questions of scale are identified including scale variability, identification of process scale, and the challenge of aggregation. Examples of scale dependence in hydrological modelling are also provided.

2.3.1. Definition of scale

Aspects of scale

There are various interrelated concepts within the definition of scale. For example; resolution, which is the size of the smallest resolvable feature; extent, which is the limit enclosing the area included; and zoom, which is the distance between the viewer and the model[2]. Zoom and extent are only independent if page (or screen) size can vary. Page

[2]In its common use, the viewer to model distance, known as zoom, is usually a synthetic distance that only exists within a computer generated visualisation

size is however not an independent characteristic of scale because it can be derived from zoom and extent. Figure 2.2 presents these aspects of scale as bar gradients, where scale can slide from left (coarse) to right (fine).

An alternative but largely analogous categorisation of scale is presented in Zhang et al. (2014) who identify three types of scale: geographic (extent), measurement (resolution), and cartographic (zoom). There are other terms that are essentially analogous to resolution such as measurement scale, grain, and support. "Support" is the integral of space or time over which a measurement is made. Many distributed variables measure a finite area, rather than a point, to produce a value.

Other concepts can be derived from the three aspects mentioned above. For instance, cardinality, which is the total number of distinguishable features, can be determined for a raster data set as extent (total area covered), divided by resolution (areal cell size). Another related concept is distribution. The distribution of a spatial dataset can refer to what is called the geometry of a point set in Chapter 3 but can also represent a form of scale that varies from dense to sparse. Dense datasets are typified by raster data models and sparse by vector data models. Judicious use of sparse distribution maximises information content for a given scale.

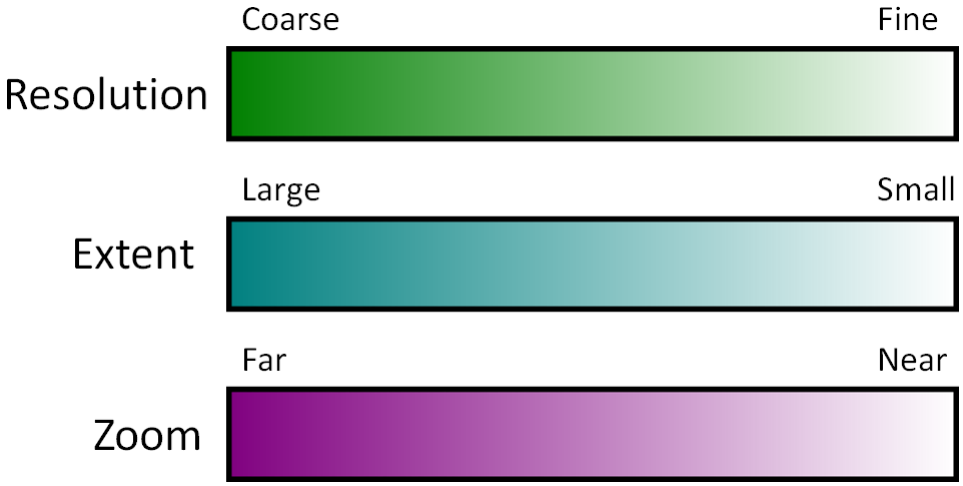


Figure 2.2.: Aspects of scale. In an appropriate scale model; resolution, extent, and zoom are aligned.

When small is large

Scale is often attached to an adjective to indicate relative size or relative resolution. For instance, both the scale of an effect and the scale of a map may be described as "small scale" or "large scale". These phrases are highly ambiguous and a cause of semantic confusion because in different contexts the phrases can have opposite meanings. A large

scale effect is typically one which affects a large area. Whereas, a large scale map is one where the ratio of map distance to real world distance is comparatively large. Therefore, a large scale map is a map of a small area in great detail. Various forms of this confusion permeate discussions of scale and resolution. For instance, what does the term "higher resolution" mean? Many would answer fine resolution but in some contexts it may refer to higher abstractions of the data: coarser representations high in a pyramid of Levels of Detail.

To avoid ambiguity, this thesis will avoid the use of the ambiguous terms large/small and high/low when discussing scale. Instead, the terms fine and coarse will be used, which are less easily confused. The exceptions to this guideline are those aspects of scale that fine and coarse do not describe well: extent or domain, which is best described with large and small; and zoom, for which near and far are more appropriate.

2.3.2. Appropriate scale

The concept scale combines the three aspects depicted in Figure 2.2 to describe a characteristic that is both subjective and relative. It is helpful to consider the question: what is appropriate scale? If each of the aspects in Figure 2.2 are thought of as ranges of possibilities and a specific scale is represented by a vertical line some distance along the interval, an appropriate scale is a scale where the vertical lines of all three aspects are aligned.

A paper map is a symbolic model of reality that has scale composed of the three aspects. If that scale is a coarse resolution dataset of a large extent at far zoom, or a fine resolution dataset of a small extent at near zoom, one would say that was an appropriate scale. However, if the map depicted a coarse resolution dataset of a small area at far zoom, one would consider that an inappropriate scale (see Figure 2.3).

Multi-scale online maps provide a way to circumvent scale limitations by storing fine resolution data over large extents but presenting realisations to map readers at a scale appropriate for the readers' zoom level (Muehlenhaus, 2014). This is a pleasing analogy of the thinking behind the surface model that is proposed in Part II.

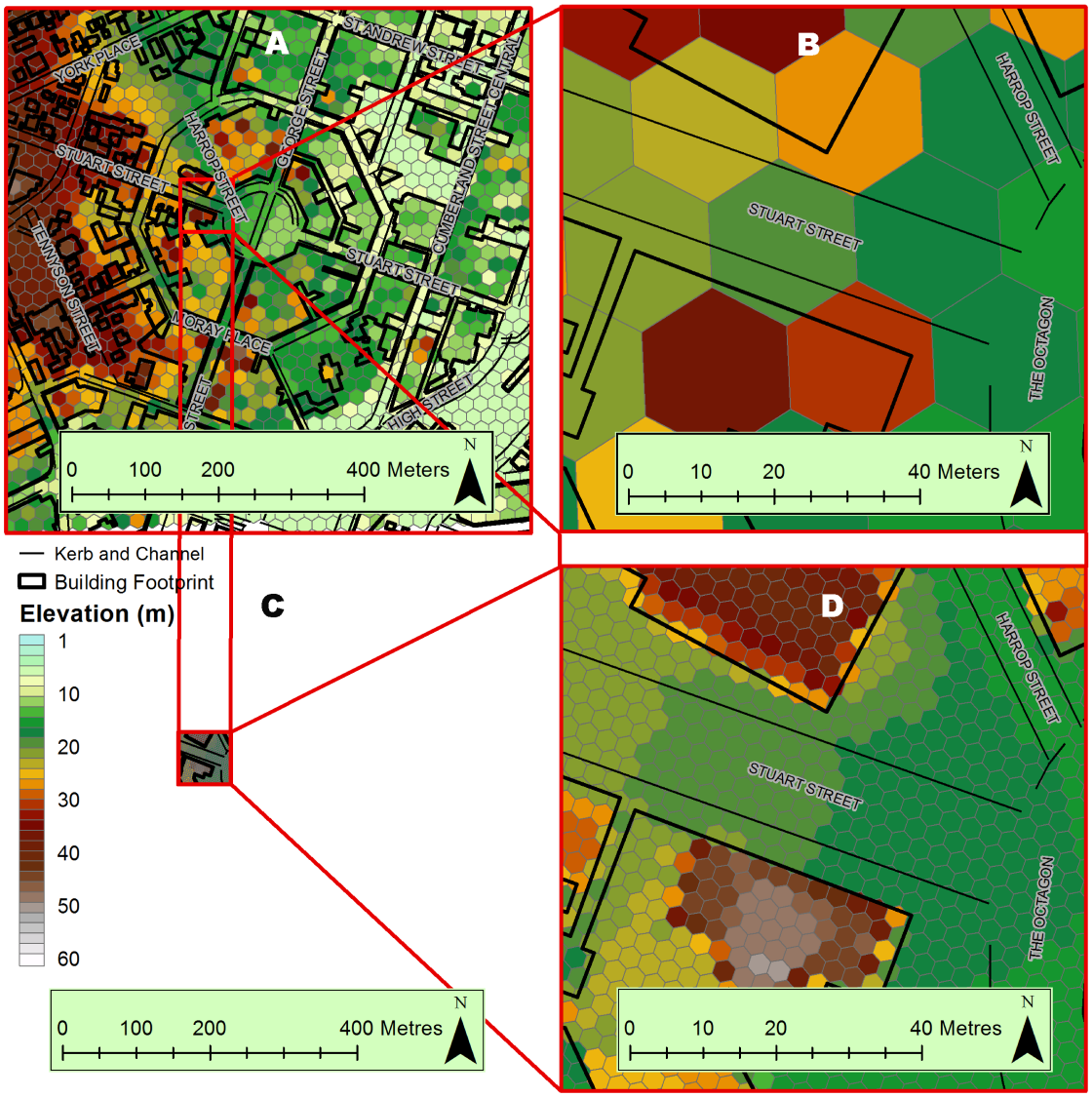


Figure 2.3.: Appropriate scale. The hexagonal elevation surface is shown with appropriate scale in A (Coarse resolution, large extent, and far zoom) and D (Fine resolution, small extent, and near zoom). All these terms are relative; the resolution of A is coarse relative to D but may be considered fine in other circumstances. B has a smaller extent than A and an appropriately nearer zoom, however, the resolution is not finer, therefore, there is little information shown and the hexagonal tessellation is out of proportion with the vector information. C has finer resolution than A and smaller extent, however the zoom level is unchanged created an irrationally small and difficult to read diagram.

2.3.3. Questions of scale

Modelling environmental processes involves complex interactions between the process being modelled and measurement scale, computational scale, and reporting scale. From these interactions, several questions of scale arise including whether variables vary with scale, how to aggregate and integrate measurements made at different scales, and how to select the appropriate scale to represent a process or effect.

Scale variability

The fractal nature of geographic phenomena causes some measurements to be affected by scale. The most well known example is the surprising conclusion that the length of a coastline is a function of the scale at which it was measured (Goodchild and Mark, 1987; Mandelbrot, 1967). Goodchild (2001) observes that a variable is scale dependent if a length is inherent in the definition of the variable. Parameters involving slope are particularly affected by measurement scale (Wang and Yin, 1998). The slope of a surface depends on the length over which it is measured. Take for example a rocky plateau; over a length of 100m it may be comparatively flat with little variation in elevation. However, over the length of a metre, the variation of large rocks and boulders becomes significant and the slope will be much steeper.

Scale dependent variables do not necessarily approach a limit with finer resolution. Looking closely at the surface of a rocky plateau may reveal a similar shape of surface variability at a finer scale, perhaps at centimetre support. Plausibly, if the fractal model holds, the same pattern may be seen again at millimetre scales and so on. Hence, all scales of a fractal phenomena are generalisations, which ignore finer scale variability. Scale dependence of terrain can be quantified by fractal dimension (a measure of the proportion of fine to coarse scale relief), semi-variance (the correlation of relief with distance), or the power spectrum (a technique of waveform analysis) (Pike and Kimberly, 2005). Alternatively, some variables may not exist at all outside certain ranges. A trivial example is that slope does not exist at 100 000km support; the spherical shape of the Earth makes such measurements meaningless.

Process scale

Use of appropriate scale is necessary but not sufficient for distributed modelling. It is also necessary to consider the scale at which a process operates in the environment. This is process scale, also known as operational, or intrinsic scale (Blöschl and Sivapalan, 1995). To optimise representative power of a model, appropriate scale must correspond with

process scale i.e. data resolution should be equal to information granularity. How to identifying process scale is one of the critical questions of scale in GISc.

One way to identify process scale is to identify the scale where there is the greatest variability in the data. The classic approach is the Geographic Variance Method (Moellering and Tobler, 1972), which identifies the scale where the sum of squared variances is greatest. The general principle is that at scales finer than the process scale, the measured variables will vary less due to the measurement scale being finer than the rate of change with space. At coarser than process scales, the averaging implicit in the larger computational scales causes reversion to the mean and reduced variability. The Geographic Variance Method assumes underlying homogeneity of the dataset rather than fractal self-similarity or significant noise at the finest scales.

Variograms are another way to identify process scale (Peckham et al., 2009). Variograms model the correlation of pairs of point measurements for a selection of lag distances. The range of a variogram, where variability stops increasing with distance, gives an upper limit for process scale.

In Part II, Level of Detail methods for modelling surfaces described in Section §3.8 are proposed as a means to identify process scale for distributed hydrological models.

The aggregation challenge

To accurately portray geographical phenomena, such as hydrology, it is necessary to understand processes that occur over a range of temporal and spatial scales. The challenge of scale is achieving a realistic description of a spatial process that accounts for spatial variability, which involves both choosing an appropriate scale and conversion of heterogeneous data inputs to this scale. Coarse scale modelling requires aggregating processes that occur at finer scales. The difficulty of aggregation depends on the degree of spatial variability of spatial processes (Famiglietti and Wood, 1994). For non-linear processes in heterogeneous environments, aggregation is very challenging.

It is a common desire to represent a phenomena at a coarser scale than that at which the measurements from which the process is being modelled were made. A modeller may desire to use point measurements to inform attributes that will be applied to a region, or to reduce the number of calculations required to find a mathematical solution, or to combine the results with processes that are modelled at a different scale. For example, comparatively fine scale hydrological models are often combined with atmospheric climate models that have a much coarser scale.

A scaling function would allow the definition of grid or element scale equations and para-

meter values on the basis of knowledge of the parameter values at finer scales, in cases of non-linear processes this is not simple or generally sound (Beven, 2001). If a system is linear, equations for processes at fine scales can be integrated over a larger specified domain to give the relationships at coarser scales (Dooge, 1986). In non-linear systems, even if mean parameters are available, it is unsafe to assume mean responses will be produced (Overgaard et al., 2006). The ramifications of generalising variables for non-linear models is illustrated in Figure 2.4. Hydrological process are largely non-linear, so issues of scale in hydrological modelling are an inescapable challenge (Kampf and Burges, 2007).

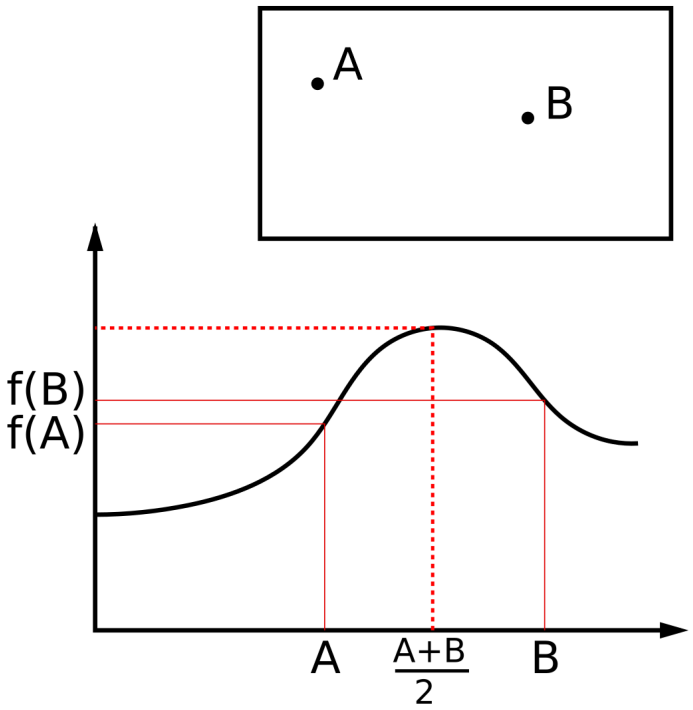


Figure 2.4.: Generalisation of non-linear functions. In this hypothetical example, a spatially distributed variable has been measured at points A and B. The measured variable contributes to the graphed non-linear model. If a model value was required for the entire rectangle it may be tempting to average the two measurements before applying the model. However, as the dotted line shows this would provide a very misleading result.

There continues to be debate about the scale at which the physically based distributed models of the Freeze and Harlan (1969) blueprint are appropriate (Beven, 1996; Kavvas, 1999; Singh and Woolhiser, 2002). The difficulties presented by the need for scaling functions to transfer processes accurately between scales has led to calls to abandon the Freeze and Harlan (1969) blueprint, either in favour of semi-lumped models like TOPMODEL (Beven and Kirkby, 1979), where lumped properties are informed by spatial analysis (rather than the process itself being modelled spatially) or for the development

of scale free governing mass and momentum conservation equations that convert the spatial gradients into fluxes across the boundary of coarse scale representative elements (Reggiani and Schellekens, 2003; Cherkauer et al., 2003; Wood, 1998). This task is made difficult by the heterogeneity of hydrological domains and difficulty of measuring mass or energy fluxes across large surfaces.

2.3.4. Scale dependence of hydrological parameters

Numerous research groups have analysed the dependence of derived hydrological parameters on spatial resolution using a number of different surface models including rectangular tessellations (Garbrecht and Martz, 1994; Li and Wong, 2009; Wang and Yin, 1998; de Sousa et al., 2006; Walker and Willgoose, 1999; Quinn et al., 1995; Gyasi-Agyei et al., 1995), hexagonal tessellations (de Sousa et al., 2006), and TIN data sets (Vivoni et al., 2005a) to name a few. For more references about the impact of DEM error and grid spacing on terrain modelling applications see Pike (2002). Not surprisingly, many studies have found that the required resolution of a hydrological model depends on the terrain. However, model resolution cannot be governed solely by desired output characteristics, precision of input data are also a critical limiting factor. Anderson et al. (2006) investigated the Resolution of LiDAR data required for various grid DEM cell resolutions.

Resolution in the vertical axis has also been studied and constrains plausible horizontal resolution. Gyasi-Agyei et al. (1995) suggest using vertical resolution where the ratio of average drop per pixel and vertical resolution is greater than unity, thus placing a lower bound on cell size. Walker and Willgoose (1999) show that reducing grid size below the vertical accuracy will result in DEM noise being mapped rather than actual information.

Shary et al. (2002) identify the need to find scale-free parameters by identifying limits as resolution increases. Processes need to be highly automated and efficient to allow recalculation on multiple scales (Lacroix et al., 2002). Variables need to be scale free in order to allow comparisons between different scale results (Shary et al., 2002). Generally, modellers are striving for the coarsest resolution possible to reduce computational workload. Coarse resolution can be sufficient in areas of complex terrain, provided the resolution is fine enough to capture the terrain (Li and Wong, 2009; Wang and Yin, 1998).

Vivoni et al. (2005a) term the effect of spatial aggregation of terrain attributes on hydrological parameters “hydrologic sensitivity” and propose a system to consider hydrological sensitivity in TIN surface creation. Florinsky and Kuryakova (2000) describe a statistical method to estimate the appropriate grid size to represent a process by maximizing correlation between outputs and landscape characteristics. Chapter 5 formalises these ideas by introducing a generalised conceptual model for capturing scale variability.

Alternatively, surfaces can be investigated for wave signals using fast Fourier analysis (Harrison and Lo, 1996). It is possible to use fast Fourier transforms to analyse scale effects but this technique is limited because it assumes stationary of signal, uses a fixed sample window, and assumes landforms can be represented by sine waves (Gallant and Hutchinson, 1997). Gallant and Hutchinson (1997) argue that wavelets are more suitable because they adjust window size as required.

This chapter began with a generic discussion of the modelling process and has explored the concepts of distributed parameters, scale, and aggregation. One of the most ubiquitous digital representations of distributed parameters is the Digital Elevation Model (DEM), which is the topic of the next chapter.

3. Digital Elevation Models

Digital Elevation Models (DEMs) are digital representations of elevation surfaces in 2.5 or 3-D reference frames. DEMs have many uses in map making, visualisation, and data analysis techniques. The DEMs frequently encountered in GIS are square grid rasters, Triangular Irregular Networks (TINs), contours, and point clouds. Of these, square grid rasters are the most common. However, these models are only a subset of the possible combinations of structures, geometries and data models that can store elevation data. Each have distinctive characteristics, which may make them more, or less, suitable for specific applications

The purpose of this chapter is to catalogue the DEMs in common use within a consistent framework. Treating the models as part of a cohesive framework as opposed to a collection of distinct types provides insight into the assumptions made in algorithms that rely on specific elevation surfaces and suggests ways that they could be generalised to other surfaces. A detailed discussion of the hydrological variables and objects generated from geomorphometric analysis of DEMs follows in Chapter 4.

Section §3.1 summarises the primary considerations of the conceptual models of DEMs. Section §3.2 outlines the ground, aerial, and orbital sources of data for DEMs and summarises techniques for validating surfaces. The remaining sections define four distinct aspects of a cohesive framework for cataloguing surface models.

1. *Structure*. Structures define the relationship between data points. Section §3.3 describes the structures used for DEMs: point sets, network graphs, and tessellations.
2. *Geometry*. Geometry describes the spatial arrangement of data points. Section §3.4 discusses and compares irregular geometries used in DEMs: ad-hoc, Voronoi diagrams and Delaunay Triangulation, irregular triangles, and boundary fitted grids. Section §3.5 discusses and compares regular geometries used in DEMs: triangular, rectangular, and hexagonal.
3. *Interpolation*. Section §3.6 discusses the interpolation methods used to produce values away from known data points, their application to resampling and restructuring, and distinguishes them from generalisation.

4. *Data model.* Section §3.7 summarises three categories of data model utilised to encode, process, and communicate elevation data: raster, vector and tree models.

Section §3.8 concludes the chapter by introducing the structures, geometries, and data models of multi-scale surface models.

3.1. Conceptual models

This section addresses three questions that must be answered to form a conceptual model for a DEM. What is elevation? What is an elevation surface? What should be included in the elevation surface? Notwithstanding the great variety of possible responses to these questions, some answers are given that correspond to common practice within the GISc community.

3.1.1. What is elevation?

An elevation is the height of a point above a reference surface known as a datum. The reference surface could be a plane, sphere, ellipsoid, or geoid model. The geoid is an equipotential surface of the Earth's gravity field, which coincides with mean sea level. A height can be defined as the difference between two elevations. Therefore an elevation is a height but heights are not necessarily elevations. Height is frequently used to denote the distance between the top of geographic feature and its base.

Hannah (2009), citing a French language paper (Heiskanen and Moritz, 1967), identifies four types of heights in common use. One that does not reference gravitational potential and three that do:

1. An ellipsoidal height is the distance between a point and an ellipsoid along a line that intersects that point and is normal to the ellipsoid. Ellipsoidal heights do not reference gravitational potential. Global Navigation Satellite System (GNSS) receivers determine ellipsoidal elevations.
2. An orthometric height is the distance along the plumb line between a point and a reference surface. Elevations for everyday use are typically orthometric heights relative to local datums derived from mean sea level as measured by continuous long term tide gauges. Mean sea level deviates from the geoid by up to $\pm$2m due to oceanographic and meteorological effects. Ellipsoidal heights can be converted to orthometric heights using a geoid model.

3. A dynamic height is the difference in gravitational potential along the plumb line between a point and a reference surface. Dynamic heights indicate the direction water will flow.
4. A normal height is the difference in normal gravity along the plumb line between a point and a reference surface. Normal heights are important to modern geodesy but are rarely used outside of the scientific community.

3.1.2. What is an elevation surface?

Most DEMs take a functional approach and consider terrain as a bi-variate scalar field. A DEM consists of a set of points with associated elevation values and interpolation rules to define the values between the points (Gerstner, 2003). By convention, x and y specify location on a horizontal plane and z or $h(x, y)$ represents the height or elevation at that location. Functional models are continuous: they have values everywhere and the values are the same regardless of the direction from which the location is approached. A functional surface model can only store one elevation value per (x, y) location. The land surface could theoretically be described as:

$$f(x, y) = z \tag{3.1}$$

However, for most surfaces what this equation would look like or how it could be determined is unclear. Instead, the land surface equation is approximated using a DEM.

Functional surface models are sometimes called 2.5-D models. An alternative approach is solid (3-D) models that represent surfaces as the two dimensional faces of three dimensional objects. Some examples of solid models are multipoint patch, voxel, and polyhedral; which are 3-D versions of vector, raster and TIN surface models respectively. The remainder of this section focuses on 2.5-D models.

3.1.3. What should be included in an elevation surface?

DEMs can be subdivided into Digital Terrain Models (DTMs) and Digital Surface Models (DSMs). DTMs represent "bare earth" with buildings, trees and other features removed. DSMs represent the highest elevation surface including features such as buildings and tree canopies, where these exist. Whether a specific feature is included in a DEM depends on both the use for which it is intended, and the scale of the DEM. The conceptual model of

a surface assumes a certain amount of abstraction: only objects which are relevant to the intended purpose of the models should be modelled. For example, a spot height that falls on a fine scale feature like a power pole should be ignored when constructing a surface model for modelling stormwater flows.

3.2. Data sources

This section surveys data sources for elevation data, divided into ground based and aerial methods (including space), followed by an introduction to surface validation. For a summary of data sources and accuracies see (Garbrecht and Martz, 2000). The data sources discussed below have distinctive characteristics that affect model results. In fact, Li and Wong (2009) demonstrated that data source is more important than resolution in determining inundation results.

For hydrological modelling, existing databases of spatial data are a good source for constructing DEMs. In addition to raster DEMs and TINs, GIS databases frequently contain useful vector data including road centre and kerb lines, catch pits, contours, streamlines, digitized maps, and Building Information Modelling. Orthophotos are visually rich sources of information and can be analysed digitally to extract various objects of hydrological significance such as the presence of structures, roads, stream networks, lakes, catch pits etcetera. A commonly available global square grid DEM GTOPO30 , has a 30” (approximately 1 km) spacing and is compiled from many maps and several raster and vector sources. If it is necessary to combine different existing elevation models, Doytsher et al. (2009) recommend using rubber sheeting methods that match objects not coordinates.

Urban surface models require a sampling of elevation values over extensive areas. Surface sampling can be random based on a regular pattern or feature driven, where the data collector deliberately selects points of topographic significance. Points of topographic significance are typically ridge lines, streamlines, changes of grade, and tops of banks. A system of break lines and spot heights is formed to capture the landscape. Feature driven elevation sampling is usually performed by ground observations, whereas aerial methods are generally based on regular sampling schemes.

3.2.1. Ground based

Ground observations were traditionally collected using theodolites, spirit levels, and measuring chains. Electronic distance measurement (EDM) has replaced chaining and modern surveyors use total station instruments that combine a digital theodolite with an EDM.

Total station measurements of topography are usually restricted to small areas due to the time required to collect the data and the requirement for inter-visibility.

Another important technology for feature driven data collection is Global Navigation Satellite System (GNSS) also known as Global Positioning System (GPS), although GPS, strictly speaking, applies only to the GNSS system operated by the United States military. GNSS data collection is quicker than total station data collection, and with careful procedures can produce high accuracy positions (<10mm horizontal, <30mm vertical) over large distances. GNSS receivers can be mounted on vehicles to expedite data collection in a semi automated procedure. GNSS surveys yield ellipsoidal heights so a geoid model is needed to combine GNSS elevation data with orthometric heights from other sources.

The most accurate form of terrestrial elevation data collection is spirit levelling. Spirit levelling only produces height differences, therefore, horizontal position must be determined independently by another method, for instance GNSS or total station. Elevation accuracy can be achieved at sub-millimetre level over hundreds of metres, although, data density is typically very low due to the time required to collect elevation by spirit level. Due to the high costs, levelling is generally only suitable for high accuracy control surveys. Ground based data collection is difficult in urban areas because there may be many land owners to consult and request entry rights from; and many obstacles to inhibit the inter-visibility required for total stations and spirit levels, and the sky visibility required for GNSS.

Ground based remote sensing devices can also collect elevation data. Terrestrial laser scanning can be tripod mounted or handheld. Handheld laser scanning can achieve sub millimetre precision over small (sub metre) areas.

Another potential source of DEM data are ground based digital photos (Dowling et al., 2009). Small high resolution DEMs can be made using structure from motion techniques popularised by Photosynth (Microsoft Corporation, 2009). Microsoft has discontinued Photosynth, however, the surface generation algorithms are available as opensource projects, such as, bundler (Snavely et al., 2006).

3.2.2. Aerial and space

Surface information can be collected remotely from conventional aircraft, unmanned drones, or spacecraft. Remote sensing can be active or passive. Active remote sensing technologies emit and receive pulses of electromagnetic radiation or sound, whereas, passive technologies utilise ambient illumination. Remotely sensed data frequently samples land surfaces using a regular sampling structure rather than concentrating data collection

on areas of increased importance. Surface models and estimates of impervious surfaces are often generated from remotely sensed data.

Traditional photogrammetry is a passive technology where stereo-pairs of orthorectified images are used to determine local elevations. Outputs from photogrammetry are frequently stored as contour surfaces or point clouds and can be further refined for surface analysis into rectangular grid DEMs or other structures. Modern Structure from Motion photogrammetry extracts 3-D surfaces from large numbers of images taken from a moving platform (Bolles et al., 1987).

LiDAR, also known as Aerial Laser Scanning (ALS), is an active data collection technology that measures surfaces based on the return time and wavelengths of pulses of ultraviolet, visible or near infra-red electromagnetic energy. LiDAR produces point clouds with a regular sampling, very fine density (1m or less), and good vertical accuracy (Hutchinson et al., 2009). LiDAR is typically more expensive than photogrammetry for a given area, however, it achieves finer resolution results and is being collected routinely in many cities.

In addition to LiDAR, Sound Navigation And Ranging (SONAR), which utilises sound waves, and Radio Detection And Ranging (radar[1]), which utilises microwave frequency electromagnetic radiation can be used to generate elevation data. SONAR is particularly useful for hydrographic data collection. Synthetic Aperture Radar (SAR) utilises the motion of the platform to capture finer images than traditional radar by increasing the effective antennae size. Interferometry Synthetic Aperture Radar (InSAR) analyses the phase differences of multiple SAR images to produce surface models or deformation surfaces. Persistent scatterer InSAR (PSI) has been used to monitor ground deformation in urban areas (Terrafirma, 2011).

Photogrammetric data from conventional aircraft provides coarser horizontal and vertical precision than Light Detection and Ranging (LiDAR), however, price advantages and large existing databases will ensure continued use. Photogrammetric capture of elevation data from unmanned drones is widely used due to the fine resolution that can be achieved and lower cost compared to LiDAR. Fixed wing drones can survey larger domains than rotor based systems but are unable to hover. An error analysis for LiDAR and automated photogrammetry is given in Höhle and Höhle (2009).

LiDAR collects multiple returns so it is possible to determine both vegetation returns and land returns (Evans and Hudak, 2007; Evans, 2003). LiDAR is qualitatively different to sparse point data with very different error profiles and techniques for realizing surface models (Hutchinson et al., 2009). LiDAR is better at detecting ground surfaces through vegetation than photogrammetry. There are many examples of studies which use LiDAR

[1]Originally RADAR but typically no longer capitalised.

data for urban surface modelling (Smith et al., 2005; MacMillan et al., 2003; Bandaragoda, 2008).

Remotely sensed elevation data can be collected from space based platforms. Space based remote sensing is very good for collecting regional and continental scale datasets. Space based platforms benefit from high temporal resolution and a consistent schedule of data capture. Historically, the resolution of surface models generated from space based platforms has been too coarse for fine scale urban drainage modelling. However, the current generation of satellites such as Worldview3 and 4 capture panchromatic imagery at 0.31m Satellite Imaging Corporation (2014). Some commonly used DEMs are derived from Advanced Space-borne Thermal Emission and Reflectance Radiometer (ASTER) and Shuttle Radar Topography Mission (STRM), which utilized InSAR.

3.2.3. Validation of DEMs

Elevation data will inevitably contain errors. Errors can be categorised as random, systematic, or blunders. Errors can be from the underlying sample data or the interpolation process. For a discussion of surface errors see Schneider (2002). It may be useful to assess the errors of derived variables rather than the elevation values; for instance, derivatives such as slope and curvature. Assessment of DEMs needs to consider shape rather than merely vertical accuracy. Wise (2000) reviews methods of assessing DEMs.

There are a range of validation techniques to estimate the error created interpolating DEMs including split-sample, cross-validation, and jackknifing (Smith et al., 2005; Evans and Hudak, 2007). In principle, these methods work by randomly sampling the surface then reinterpolating from the sample and comparing surfaces; or by ignoring some points in the interpolation and using those points for the validation. Systematic errors may be identified by locating unexpected variations of the spatial structure, on the assumption that the surface being modelled is relatively smooth. Such variations can be investigated by elevation histograms, spectral analysis, semi-variograms, and principal component analysis (Wise, 2000).

The error of a DEM can be quantified by checking a random sample of heights against known heights from a more accurate source. The error can then be reported as a Root Mean Square Error (RMSE), calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} d_i^2}{n}} \tag{3.2}$$

where d is the height difference between the DEM and test point, and n is the number of test points. However RMSE is sensitive to outliers, does not indicate the spatial distribution of errors, gives no indication as to the source of errors, may rely on a small number of test points, and relies on the existence of appropriate higher accuracy data for comparison. In addition, an object heighted in the reference data may have been intentionally ignored in the surface model due to the scale of the conceptual model. Comparing a point reading of elevation with a value in a DEM that is representative of an area is an example of the issues of scale and aggregation discussed in Section §2.3.

Fisher (1998) compares DEM values to known higher accuracy elevations and uses values and spatial variation of errors in a DEM as input to Monte-Carlo modelling. Smith et al. (2005) compare the gridded DEM with raw ALS data. Erdogan (2009) examines the distribution and magnitude of elevation errors in fine scale DEMs. They find errors are concentrated in areas of steep terrain and relate errors to terrain using ordinary and geographically weighted least squares (Erdogan, 2009). Höhle and Höhle (2009) present methods to assess the accuracy of bare Earth DEMs. They identify median, normalized median absolute deviation, and sample quartiles as appropriate accuracy assessment measures and provide guidelines for the appropriate sample size.

3.3. Structures

Given a set of n points in 2-D space with associated height values (h) that vary spatially:

$$\left(x_i, y_i, h\left(x_i, y_i\right)\right), i \in \mathbb{Z} : 1 \leq i \leq n \tag{3.3}$$

there are three distinct structures that define the relationship between the points to create a surface model. These structures are described below and illustrated in Figure 3.1.

1. Point set models consider the points as separate entities and treat the remaining space as a void.
2. Network graph models connect the data points using line segments to form a network of points and treat the remaining space as a void.
3. Tessellated models consider the data points as the centres of regions of space that collectively cover the entire space without gaps or overlaps.

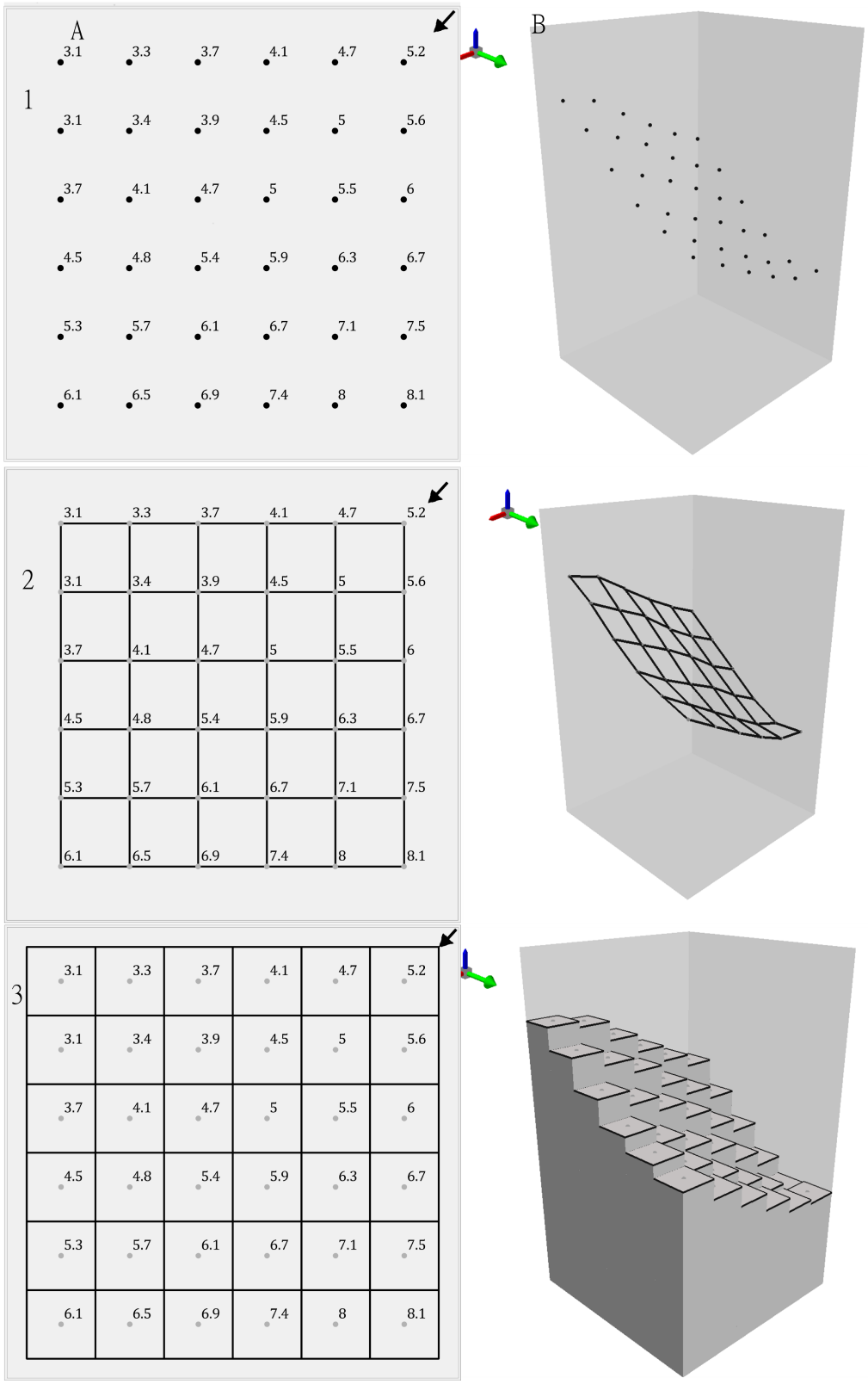


Figure 3.1.: Surface model structures. Column A shows a 2-D plan of (1) a point set, (2) a network graph, and (3) a tessellation of elevation values. Column B shows 3-D perspective representations of the same structures. The direction of view of the perspective images is shown by the arrow in the plan views.

The three structures may be combined with an interpolation method to estimate values between data points using defined mathematical functions (see Subsection 3.6.1). In this discussion, "mesh" is used as a general term for both networks and tessellations. For a summary of structures of surface models Gold (2009).

Goodchild et al. (2007) calls the components of spatial discretisation geo-atoms and defines them as a property, location tuple. Geo-atoms the basis of a general theory of geographic representation, including geo-fields and geo-objects. The structures described below are forms of geo-fields. Geo-fields are grouped by a common but varying property. A surface discretisation is a collection or integration of geo-atoms across a domain that vary in the value of some characteristic.

Goodchild et al. (2007) further characterises six common 2D surface discretisations; polygons, TINs, grids, irregular points, regular points, and contours. However, Goodchild et al. (2007) do not analyse the qualities that distinguish these different discretisation: structure, geometry, interpolation, and data model.

3.3.1. Point sets

A point set DEM consists of a collection of geo-atoms located in a two or three dimensional reference frame. There are no specified relationships between the points of a point set, nevertheless, relationships are often perceived by the viewer, based on relative proximity (Gold, 2009). Each node in a point set DEM has a single value representing the elevation at that point. Point sets are the primary building blocks for other surface models. Point sets can be used as nodes in a network graph, as generator points in a tessellation, or as the sample points from which a spatial function is formed. Point sets may have regular or irregular geometries (see Section §3.4 and Section §3.5). Regular point sets can be efficiently stored using a raster format, however, irregular point sets are more easily stored using the vector data model (see Section §3.7).

A point set DEM is known as a point cloud. Point clouds typically consist of a large number of points in a 3-D reference frame. Point clouds are the initial product of most laser scanning systems, including airborne LiDAR, terrestrial laser scanning, and hand-held scanners (see Section §3.2). Point clouds are frequently further processed to form grid rasters, triangulations, or 3-D objects. Point clouds can be combined with break-lines before generating grid or triangulation surfaces to produce better results. Contours are characterised below as an example of a network graph; however, they can also be conceptualised as a point set of lines where each line is an independent object with a single value.

3.3.2. Network graphs

A network graph connects the nodes of a point set into a network using line segments (often called edges) to indicate which nodes are connected. The edges are generally straight lines but can also be curved. The topographic surface is created by assigning(x, y, z) values to each node. For a DEM, the (x, y) values typically define the position of the node in the network graph. Network graphs are node centred meshes as opposed to cell centred meshes, which are tessellations. Network graphs are the object of study of a branch of discrete mathematics known as graph theory (Biggs et al., 1998).

A network graph can be formed from a regular or irregular point set. Each node may have a consistent number of connections or the number of connections may vary between nodes. The number of cells connected to a given node is known as the connectivity or valence of the node. If a graph contains loops it is a cyclic graph. Graphs without loops are acyclic. The edges of graphs may be assigned direction to form a directed graph. Nodes and edges may be weighted with a cost value. If any two vertices in an undirected graph are connected by exactly one path it is a tree graph. If a directed acyclic graph has only one node from which all nodes can be reached it is a directed tree graph. A forest is a disjoint union of trees.

Network graphs can be tailored to describe the important points of topography, for instance, Morse-smale complexes (Comić et al., 2005; Danovaro et al., 2006, 2003). Both the nodes and the edges of a network graph can be positioned to follow topographic features (Vivoni et al., 2004).

Two commonly used network graph surface models are Triangular Irregular Networks (TINs) and Contours. A TIN surface model consists of nodes connected by lines that form the edges of adjacent triangles. TINs are generally formed from irregular point sets, however, Triangular Regular Networks are also possible, for instance, Agüero et al. (2003) form a Triangular Regular Network using structured total least squares. The triangular facets formed by the edges of a TIN can be treated as a plane to form a functional surface (see Subsection 3.6.1).

Contours represent elevation using lines, which may be curved. Each contour line is the intersection of the surface being represented with a plane of equal elevation. For large areas the plane is a curved portion of an ellipsoid or an equipotential surface of the Earth's gravity field. Contour based elevation models were used in the 18th century. Dupain-Triel compiled a contour map of France in 1791 (Imhof, 2007). Today, large databases of both digital and analogue contours form an important repository of elevation information.

Contours are visually rich and easy to interpret for experienced users. Contours are frequently used as the source data for other forms of DEMs, particularly rectangular grid

DEMs (Hutchinson, 1989). Elevation contours are generally considered functional surface models as an interpolation along lines of steepest descent can be assumed. Contour surfaces are commonly used for manual hydrological analysis. However, contours would need to have very fine resolution to capture urban drainage structures unless ancillary data are used. Contours have also been used for automated hydrological analysis. For instance, Moore et al. (1988) show contours can be useful to divide areas into consistent hydrological areas.

The flow paths or slope lines formed of lines orthogonal to contour lines form a network graph surface model in themselves. Flow paths represent lines of steepest descent and are useful in hydrological analysis. Oloughlin (1986) describes methods to determine flow paths from digitized contours. It is possible to form a cyclic network graph, which is also useful for hydrology, consisting of quadrilaterals and triangles formed by the intersections of slope lines and contour lines (Pike and Kimberly, 2005; Orlandini and Moretti, 2009). Triangles occur at peaks.

3.3.3. Tessellations

A tessellation or tiling subdivides space into tiles. Every point in space is assigned to one and only one tile except for edges, which are the intersection between two tiles; and vertices, which are the intersections of three or more tiles. A tessellation of 2-D space is called a tessellation of the plane. The tessellations most commonly used in GIS are square grids, Voronoi diagrams, and various ad-hoc tessellations. In addition to hydrological flows, tessellated surfaces have been utilised to model a diverse range of flows including fire (Hernández Encinas et al., 2007), population movement (Holland et al., 2007), and cosmic flux (Romano-Díaz and Van De Weygaert, 2007). Tessellations are also useful for indexing spatial data.

Tessellations are cell centred meshes as opposed to node centred network graphs. Tiles are sometimes called cells or pixels. Although the following nuances are not universally followed, the term cell suggests that there is some interaction between the cell and its surroundings and a central point associated with the cell. The term pixel suggests that the tile is part of a regular tessellation. Tilings cover infinite space, however, DEMs are usually finite in extent. Therefore, tilings utilised for DEMs must be bounded. Tiles that are adjacent to the perimeter of a bounded tessellation are edge cells and often require special handling in spatial analysis procedures.

The shapes of tiles can be compared by the number and length of their sides, and by the magnitude of their internal angles. A regular polygon is a polygon formed of edges the

same length and internal angles of the same measure. Two tiles are identical if and only if they have have the same number of sides, the same internal angles, the same length of sides, and the same position. The shapes of tiles can be classified as equivalent, congruent, or similar based on which of the transformations translation, rotation, reflection, and scaling are required to make them identical. If two tiles can be made identical by translation alone they are equivalent. If they can be made equivalent by rotation and/or reflection they are congruent. If they can be made congruent by scaling they are similar. If two tiles can be made identical by translation and/or scaling they have the same orientation, for examples see Figure 3.2.

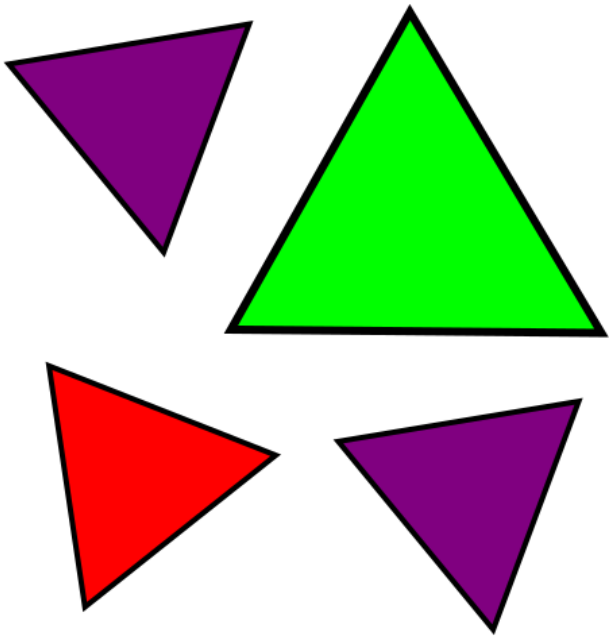


Figure 3.2.: The purple triangles are equivalent. The purple and red triangles are congruent. The green triangle is similar to both the purple and red triangles.

Tessellations can be characterised by the combination of tiles intersecting at each vertex. Connectivity (called valence in de Floriani et al. (2005)) is a measure of the number of tiles intersecting at each vertex. Connectivity is also relevant to network graphs. Tessellations that form a repeating pattern are known as periodic tessellations. All periodic [2] tessellations can be assigned to one of 17 wallpaper groups (Fedorov, 1891; Grunbaum and Shephard, 1977).

The neighbourhood of a tile is the set of tiles that share at least one vertex with that tile. Adjacency indicates the spatial relationship between neighbouring tiles. Depending on the geometry of the tessellation, all the neighbours of a tile may be adjacent across an edge or some may be adjacent across a vertex. If all the edges of a tessellation are

[2]Mathematicians also consider colour in the period of tilings. However, colour is ignored here.

the linear boundary of two or fewer tiles, it is a conforming or edge-to-edge tessellation ((de Floriani et al., 2005).

The distance of a neighbour is defined as the distance between centroids of the tiles. The neighbours of a tile may all be the same distance away or their distances may vary. The way that a tessellation packs space affects the stability and the density of the tessellation for a given cell to cell distance of tessellation.

It may be possible to decompose or aggregate the tiles of a tessellation of the plane into other shapes that also tessellate the plane. A tessellation is self-similar if tiles can be aggregated or decomposed into a shape that is similar to the initial shape.

For every tessellation, there is a dual network graph that can be formed by connecting the centroids of the cells with the centroids of the cells that share a linear boundary. Voronoi diagrams and their dual Delaunay Triangulations are described in more detail in Subsection 3.4.2. A different graph can be generated from a tessellation using the vertices and edges of the tiling. This alternative graph is what Winter and Frank (2000) call the skeleton of the raster. Conversely, a tessellation can be formed from a network graph using the cells formed by the edges of the network, and defining a value for a cell.

3.4. Irregular Geometries

The geometry of a surface model describes the spatial arrangement of data points within the surface. An irregular tessellation is a space-filling configuration of polygons of varied shape and size. Irregular arrays can be used for finite volume and finite element analysis (Mark, 1988). This subsection describes the characteristics of four irregular tessellations: ad-hoc, Voronoi diagrams, Triangular Irregular Networks (TINs), and boundary fitted grids; the following section discusses three regular geometries: triangular, rectangular, and hexagonal. For each geometry, the shape refers to the Voronoi cell created from the data point rather than the shape created by the edges of a network graph.

An irregular mesh may have an irregular distribution of points but regular or semi-regular connectivity; for instance, a TIN may consist of triangles of varying shape and size but there are always 6 triangles meeting at each vertex. An irregular mesh with regular valence is called a semi-regular mesh (de Floriani et al., 2005). Semi-regular meshes often contain some extraordinary nodes, which have a different valency. The distribution of the points in an irregular mesh may be irregular but determined by a mathematical formula as is the case for Boundary Fitted Grids or they may be ad-hoc constructions.

When addressing problems that involve computation on a regular point set, a trade off occurs between reducing the number of cells to avoid excessive computational time and

increasing the number of cells for increased accuracy (Jin and Wu, 1997). Using a fine scale surface over the entire domain may be computationally infeasible or unwarranted by the available data (Kampf and Burges, 2007). For hydrological problems, a finer spatial resolution is required in some areas to resolve flow fields, for instance, in areas of converging flow. Irregular geometries resolve this trade off by permitting greater density in areas that require it. The greater flexibility of irregular geometry comes at the expense of more complicated supporting structures and corresponding algorithms (Gerstner, 2003). Irregular geometries are normally associated with explicit data structures.

3.4.1. Ad-hoc

In an ad-hoc geometry, not only are the points not regularly distributed but the edges of the network graph or tessellation are not created in accordance with a set of rules but arbitrarily to suit on an ad-hoc basis. A regular point set could also be used to generate ad-hoc tessellations by arbitrarily positioning the edges. Also an ad-hoc network graph can be generated from a regular point set by arbitrarily connecting nodes to form a network.

3.4.2. Voronoi diagrams and Delaunay Triangulation

A tessellation of particular importance to spatial science is the Voronoi diagram, also known as Thiessen polygons, proximal polygons or Dirichlet cells. Using a 2-D example, given a set of points or "generators" in 2-D space, a Voronoi diagram is formed by assigning an area to each point such that everywhere in that area is closer to that point than any other point in the set of generators (see Figure 3.3). If the generators are regularly spaced, the resulting Voronoi diagram may be degenerate, meaning some Voronoi vertices are shared by more than three polygons. Gold (2009) characterises Voronoi diagrams as a spatial model that unifies the concepts of objects and fields by answering the question: what is closest to here? In contrast to the question: what is here?

The dual of the Voronoi diagram is the Delaunay Triangulation. Voronoi diagrams and their dual the Delaunay Triangulation have useful properties and are implemented in numerous applications. Some GIS algorithms utilise both the tessellation and dual network graph (Gold, 2009). Both regular and irregular point sets or the nodes of a network graph may be used as generators to form a Voronoi tessellation or connected in a unique way to form a Delaunay Triangulation. The Voronoi diagram is unique for any set of generators.

Okabe et al. (1992) provide mathematical definitions of the Voronoi polygon based on the nearest generator method on a plane and extends the definition to higher dimensions

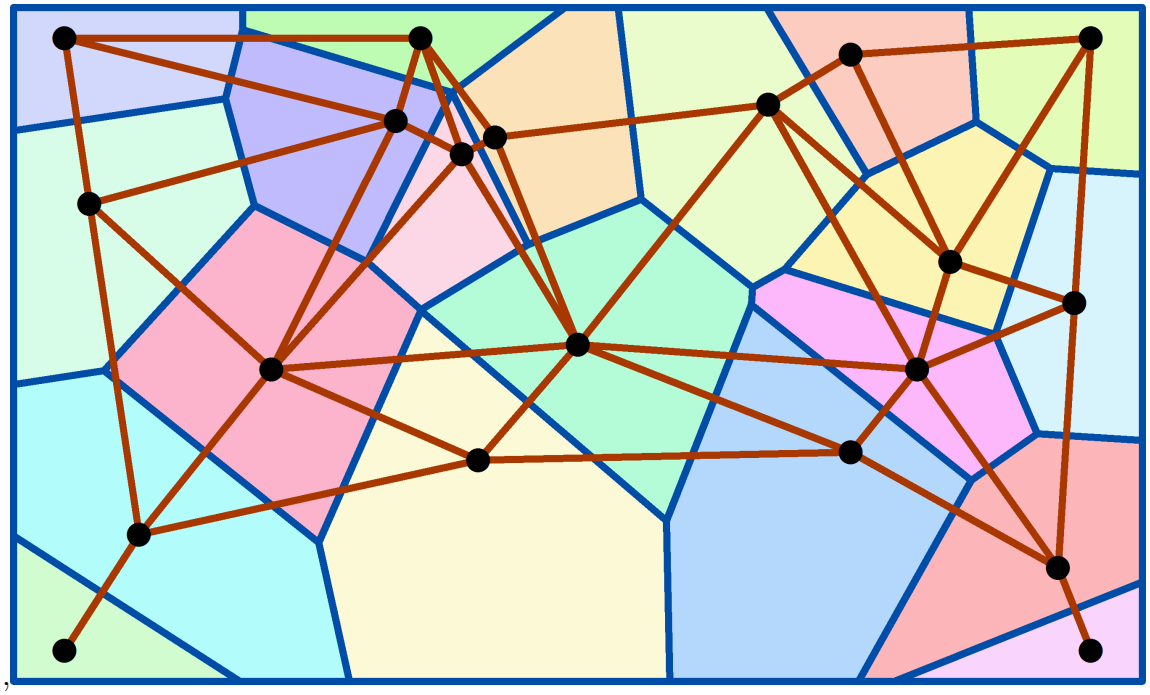


Figure 3.3.: A bounded Voronoi tessellation generated from the black generator points. The brown lines are the dual Delaunay Triangulation. The blue lines are the skeleton network graph.

using the intersecting planes definition. In an ordinary Voronoi diagram, each Voronoi polygon is associated with a single generator, the interior of each polygon encompasses all points nearer to its generator than all other generators, and the edges of a Voronoi diagram are equidistant between at least two points. In an open Voronoi set the borders are not part of the subset. In a closed one they are. Outer Voronoi diagram edges are assumed to extend to infinity. Augmented geometric graphs bound the whole area with a circle whose arcs close off outer polygons.

Voronoi diagrams were first described explicitly by Dirichlet in 1850 in two and three dimensions. Voronoi generalized the tessellations to m-dimensions in 1908. Both Dirichlet and Voronoi dealt with regularly spaced points. Delaunay is credited with developing Voronoi polygons' dual, Delaunay Triangulations. The useful properties of Voronoi polygons saw them used frequently in a range of natural and social sciences including frequent rediscoveries. However, the lack of efficient computing methods restricted their use until the mid-seventies. A detailed history of Voronoi diagrams and their dual the Delaunay tessellation is given in (Okabe et al., 1992).

Okabe et al. (1992) define the Delaunay tessellation provided the set of generator points is not co-linear. This is done by joining the generators that share a common Voronoi edge. If this consists of only triangles, it is a triangulation otherwise it is a pretriangulation.

Delaunay tessellations and Voronoi diagrams have a number of distinctive properties. For instance, circum-circles formed by the nodes of Delaunay triangles contain no other nodes. Lexicographically maximum triangulation theorem states that Delaunay Triangulations minimise variance in edge length and maximise the largest minimum angles in each triangle (Okabe et al., 1992). Therefore, considering the possible triangulations of a point set of nodes, the triangles of a Delaunay Triangulation are the closest to equilateral, with fewer long thin triangles.

The Naive Method to generate a Voronoi diagram is to draw lines from each generator point to its nearest neighbours, divide each of these lines into two equal parts with a perpendicular line and then join the perpendicular bisectors to create a polygon. Voronoi diagrams can be developed from their generator points by a number of more efficient methods, including the divide and conquer method (Shamos, 1975), a refined incremental method (Ohya et al., 1984) and the "dream transformation" plane sweep method (Fortune, 1987).

Voronoi diagrams have been generalized in a number of ways including weighted, higher order (Voronoi with more than one point for each cell), farthest point, Voronoi of various lines and arcs, and Voronoi on shapes. For details and references see Okabe et al. (1992). Recent developments in Voronoi diagrams include a generalization of constrained Voronoi diagrams into higher order constrained tessellations (Silveira and van Oostrum, 2007), a fractal Voronoi surface (Dobrowolski, 2007), application of Voronoi for modelling three dimensional fields (Ledoux and Gold, 2008), and an application of Voronoi diagrams to optimal path planning (Priyadarshi, 2007).

3.4.3. Triangular Irregular Networks

Triangulated Irregular Networks (TINs) are especially useful for terrains. As the name suggests, TINs are network graphs consisting of irregular triangular cells. TINs are generally but not always based on Delaunay Triangulations. TINs are useful because they are variable density, can be constructed to fit all data points, can be adapted to conform to known linear and point features, are edge-to-edge conforming, can readily be extended to a functional surface by fitting planes to the triangular facets, and easily support the addition or removal of nodes.

Hydrological models on TIN structures support variable resolution (Liu and Snoeyink, 2005). While TIN structures are not generally adaptive to hydrological properties, Vivoni et al. (2004) describe a method of forming hydrologically adaptive TIN surfaces, using a preliminary raster analysis to define regions of similarity.

Triangles are the 2-simplex. Therefore, triangular meshes are the simplex mesh of 2-D space. A k-simplex is the smallest convex set containing $k + 1$ points, where k is the number of dimensions of the reference frame. A k-simplex has $k + 1$ boundaries which are known as k-facets, which are $k - 1$-simplexes de Floriani et al. (2005). The 3-simplex is a tetrahedron. k-simplexes are the building blocks of many Level of Detail models (see Section §3.8) and also generalise 2-D TIN spatial analysis methods to higher dimensions.

3.4.4. Boundary Fitted Grids

A compromise between regular and irregular grids are boundary fitted grids (BFGs) (Jin and Wu, 1997), which consist of a regular array of rows and columns, which is applicable to raster data storage but the mesh is distorted to conform to the outer boundary of the study area. Zegeling (2004) describes tensor product adaptive grids, which also utilise a regular approach to define an adaptive grid. BFGs are quadrilateral network graphs with curvilinear non orthogonal coordinates that can be mapped onto an orthogonal grid in computational space. The distribution of interior grid points is a function of boundary grid points (Thompson et al., 1985). Boundary fitted grids are widely used in aerospace and computational fluid dynamics and are well suited to finite difference numerical schemes (Hoffmann and Chiang, 1993; Thompson et al., 1985).

The area for each cell in a BFG and the distance between adjacent cells varies. Therefore, these variables need to be stored for each grid cell. However, if there are a large number of parameters in the spatial process being modelled, the extra parameters that need to be stored to describe geometry of the cells are compensated for by avoiding cells located outside the study area, assuming that the area of interest is not rectangular (Jin and Wu, 1997).

Another benefit of the BFG approach is that it is possible to construct the grid to conform to linear features within the study area by subdividing the area at the linear feature, creating a boundary from the linear feature. This is clearly applicable to river basins. It is less obvious how such a process would be undertaken in urban areas due to large number of features that need to be modelled. However, roads and buildings are features that could be modelled in this fashion.

3.5. Regular Geometries

If all tiles in a tessellation have congruent shape it is a regular tiling. A semi-regular tiling contains incongruent shapes but the combination of tiles at each vertex is the same.

There are only three regular tilings of the plane that can be formed from regular polygons. They are the tilings formed by equilateral triangles, squares and hexagons (see below). There are eight semi-regular tilings that can be formed with regular polygons. These are known as the Archimedean tessellations (Grunbaum and Shephard, 1977).

A regular geometry in 2-D space can be described by two 2×1 basis vectors $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{R}^2$. The distances and directions between the points of a regular geometry can be derived from the basis vectors. Condat et al. (2008) showed that with the basis vectors grouped in a 2×2 matrix $\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 \end{bmatrix}$, the coordinates of all points in the point set is:

$$\mathbf{Rk}, \mathbf{k} \in \mathbb{Z}^{\mathbf{2}} \tag{3.4}$$

A regular tessellation can be characterised by its Schläfli symbol, which is also used to describe polygons and regular polytopes. Schläfli symbol indicates the number of tiles that intersect at each vertex (connectivity) and the number of vertices in each of those tiles in the form $\{c, v\}$ where c is the connectivity and v is the number of vertices. Semi regular polygons can be characterised by their vertex configuration, which lists the number of vertices of each tile that intersects at each vertex (Grunbaum and Shephard, 1977) in the form $(v_i, v_{i+1}, ..., v_n)$ where v is the number of vertices in tile i and n is the number of tiles intersecting at a vertex. The vertex configuration can be shortened to (v^n) if v is the same for all tiles intersecting at the vertex. The vertex configuration can also be nested to represent more variable periodic tessellations.

Surface models with a regular geometry have the benefit that you do not need to explicitly store (x, y) data for every point as position is implied. Therefore, regular arrays are more efficient for data storage. Topology is also implicit because the connectivity and adjacency can be easily inferred because, except for edge tiles, neighbourhoods are consistent between tiles. Consistent neighbourhoods also simplify neighbourhood operations. However, surface models with regular geometries introduce geometric bias to some spatial analysis including population dispersion models (Holland et al., 2007) and hydrological flow direction algorithms (Tarboton, 1997).

3.5.1. Regular Triangles

Regular equilateral triangular tilings have vertex configuration (3^6), Schläfli symbol $\{3, 6\}$. The dual graph of an equilateral triangle tessellation forms hexagonal tiles. $\mathbf{R}_{tri}$ can be characterised as two vectors $\begin{bmatrix} \sqrt{3}/2 \\ 1/2 \end{bmatrix}$, and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$, which each represent the direction and spacing of an infinite set of collinear points, except every third point is omitted.

An equilateral triangle point set is characterised by the matrix:

$$\mathbf{R}_{tri} = \begin{bmatrix} \sqrt{3}/2 & 0 \\ 1/2 & 1 \end{bmatrix} \tag{3.5}$$

The coordinates of all points in the point set can be found with:

$$\mathbf{R}_{tri}\mathbf{k}, \mathbf{k} \in \mathbb{Z}^2 \nmid 3 \tag{3.6}$$

The integers that are divisible by three are excluded to recreate the periodic variance in spacing. Combining both series of points generates an infinite array of triangular generators in 2-Dimensions.

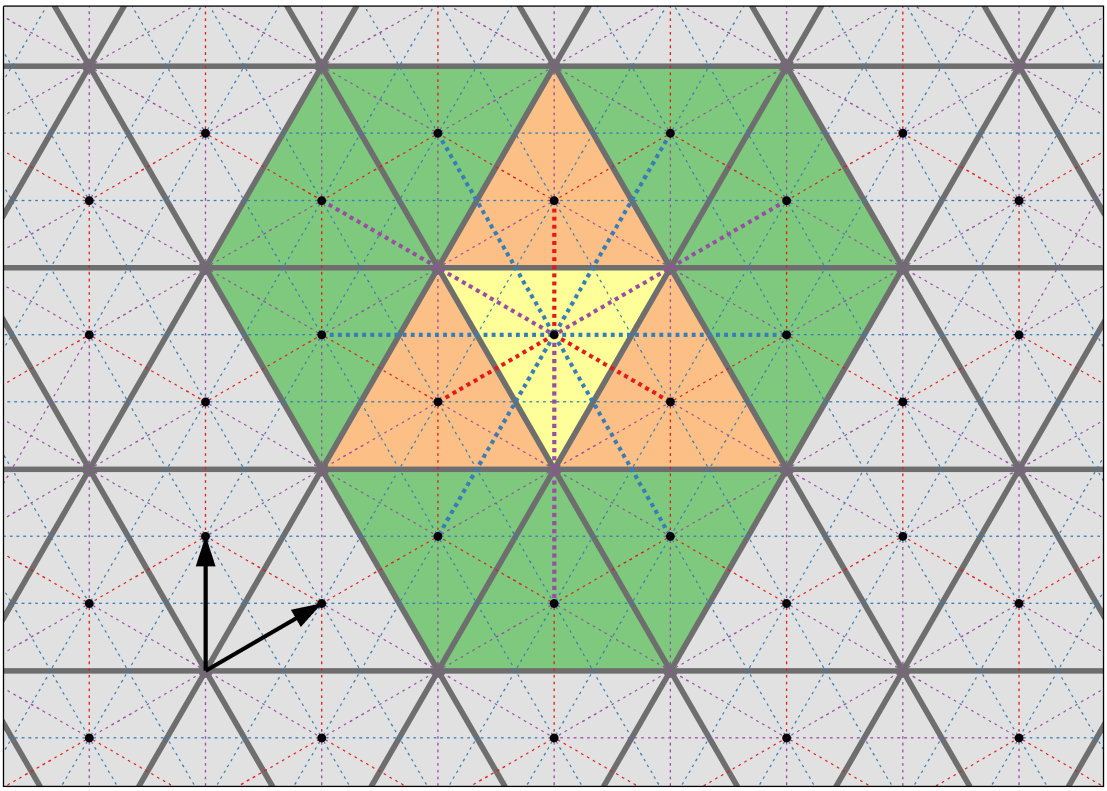


Figure 3.4.: The geometry of regular triangular tessellation. The black dots are the tile generators. The grey lines are the perimeter of the triangular cells. The black arrows represent the basis vectors of $\mathbf{R}_{tri}$. The neighbourhood of the yellow cell is coloured: the point neighbours are green, and the line neighbours are orange. The dual network is shown in dashed lines. The network lines associated with the yellow cell are shown thicker. The different length network lines are coloured differently. The network lines associated with the line neighbours (red) form the skeleton of a hexagonal array.

The neighbourhood geometry of a regular triangular tessellation is shown in Figure 3.4. There are two different orientations of the triangles in an equilateral triangular tessellation.

The neighbourhoods of triangles are not consistent. Triangles have three edge adjacent and nine point adjacent neighbours. There are three different adjacency distances in a triangular neighbourhood. The inconsistent neighbourhood of triangular tessellations complicates neighbourhood operations.

There are five variations of regular tessellations that can be formed with triangles: equilateral, isosceles, right angle, and two scalene triangle variations. Tessellations consisting of, equilateral, isosceles, and 30-60 right angle triangles can form self-similar hierarchies. The self similar agglomerations are formed from n^2 tiles, where n is any integer > 0.

The length (ω) of the projection of an equilateral triangle on a line in direction (Θ) varies according to;

$$\omega = h\left(\cos\left((\Theta \% \pi/3) - \pi/6\right)\right) \tag{3.7}$$

where h is the height of the equilateral triangular array element whose base is aligned with the y-axis.

3.5.2. Regular Rectangles

Two dimensional space can be tessellated into rectangles. Conforming regular rectangular geometries have (4^4) vertex configuration. Square grids have Schläfli symbol $\{4, 4\}$. The dual network graph of a rectangular tessellation forms rectangular tiles translated by half a cell size in x and y (see Figure 3.5).

A conforming rectangular point set is characterised by the matrix:

$$\mathbf{R}_{rec} = \begin{bmatrix} m & 0 \\ 0 & n \end{bmatrix} \{m, n \in \mathbb{R} : m, n > 0\} \tag{3.8}$$

If $m = n$, it is a square point set. The coordinates of all points in the point set can be found by :

$$\mathbf{R}_{rec}\mathbf{k}, \mathbf{k} \in \mathbb{Z}^2 \tag{3.9}$$

The orientations of the rectangles in a regular rectangular tessellation are consistent. However, rectangles have four edge adjacent and four point adjacent neighbours. There are two different adjacency distances in a rectangular neighbourhood. Some neighbourhood algorithms on rectangular arrays consider only the four edge adjacent neighbours. This

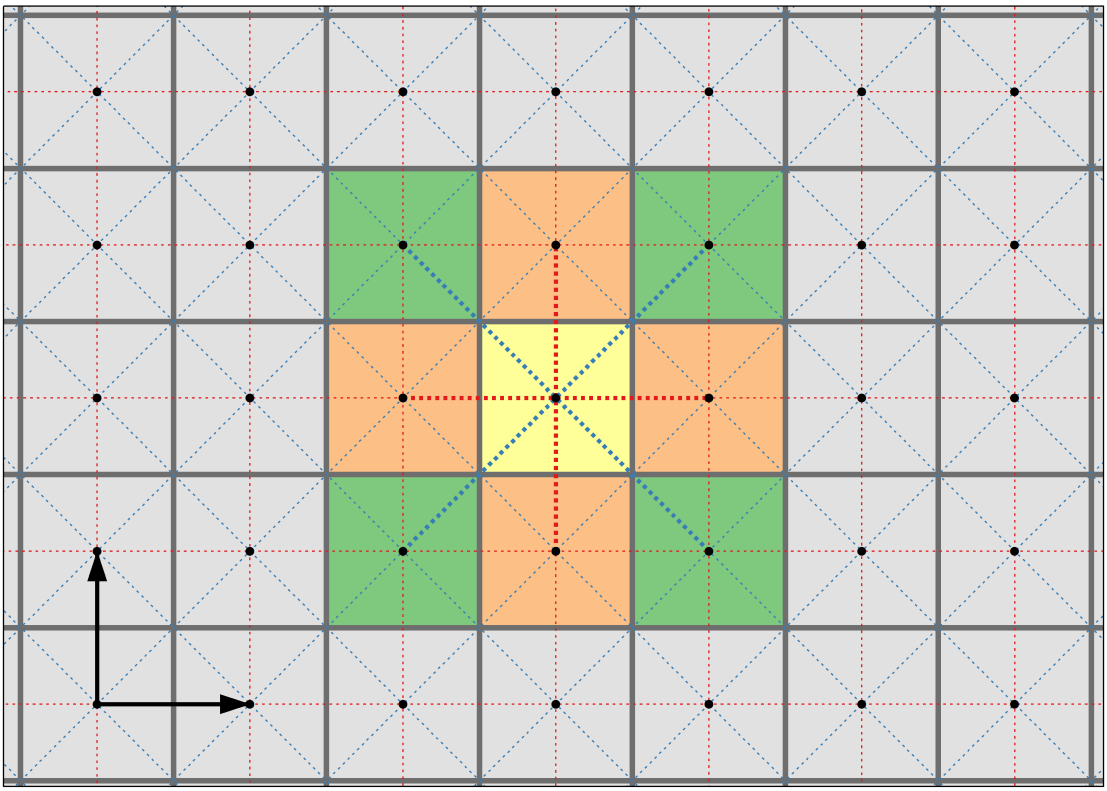


Figure 3.5.: : The geometry of regular rectangular tessellation. The black dots are the tile generators. The grey lines are the perimeter of the rectangular cells. The black arrows represent the basis vectors of $\mathbf{R}_{rec}$. The neighbourhood of the yellow cell is coloured: the point neighbours are green, and the line neighbours are orange. The dual network is shown in dashed lines. The network lines associated with the yellow cell are shown thicker. The different length network lines are coloured differently. The network lines associated with the line neighbours (red) form the skeleton of a translated rectangular array.

is known as rook's case neighbourhood. Alternatively, some algorithms use the queen's case neighbourhood, which considers all eight neighbours. However in the queen's case neighbourhood algorithms often need to treat the diagonal neighbours differently from how the rook's case neighbours are treated; this disparity is known as the connectivity problem (Golay, 1969). The connectivity problem is discussed in the context of the geomorphometric parameter flow direction in Subsection 4.3.1.

Rectangular grids are self similar. Any agglomeration of cells is self similar provided the number of cells in the x-dimension is the same as the number of cells in the y-dimension, i.e. self similar agglomerations are formed from n^2 tiles, where n is any integer > 0. Rectangular grids can be decomposed into isosceles triangles or right angle triangles.

The length of a cross section of a rectangle varies depending on the direction of transit. For a square, the cross section varies from 1 to $\sqrt{2}$ multiplied by the cell size. More

generally the cross section of a cell (ω) in a rectangular mesh can be computed as:

$$\omega = |\cos(\Theta)|\Delta x + |\sin(\Theta)|\Delta y \tag{3.10}$$

where Θ is the direction of transit referenced to the x-axis and Δx and Δy are the edge lengths of a rectangular element aligned with the x and y axes.

Rectangular grids can be aligned with Cartesian axes which makes indexing simple and facilitates alignment with display monitors and data collectors that use rectangular grids. Large grids can be processed using data tiling (McCormack and Hogg, 1997). Brick bond tessellations are a non-conforming variation of rectangular arrays; vertices occur part way along the perimeter of a neighbouring tile. Brick bond tessellations do not exhibit self similarity.

A DEM composed of a regular array of rectangles, each with a single elevation, was proposed by Miller and Laflamme (1958). Widespread availability of rectangular grid DEMs first occurred in the 1980s. This form of DEM is so ubiquitous it is frequently called DEM without further explanation. Rectangular grid DEMs are the most frequently used datasets in geomorphometry (Hengl et al., 2009).

3.5.3. Regular Hexagons

Two dimensional space can be tessellated with hexagons (see Figure 3.6). Hexagonal tessellations have (6^3) vertex configuration and Schläfli symbol $\{6, 3\}$. The dual network graph of a regular hexagon tessellation forms equilateral triangular tiles. For a general summary of hexagonal arrays, see Middleton and Sivaswamy (2005a).

A hexagonal point set is characterised by :

$$\mathbf{R}_{hex} = \sqrt{\frac{2}{\sqrt{3}}} \begin{bmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{bmatrix} \tag{3.11}$$

The coordinates of all points in a hexagonal point set can be found by:

$$\mathbf{R}_{hex}\mathbf{k}, \mathbf{k} \in \mathbb{Z}^2 \tag{3.12}$$

All tiles in a hexagonal point set have the same orientation. The neighbourhoods of hexagonal tessellations are consistent. All neighbours are adjacent across edges and are

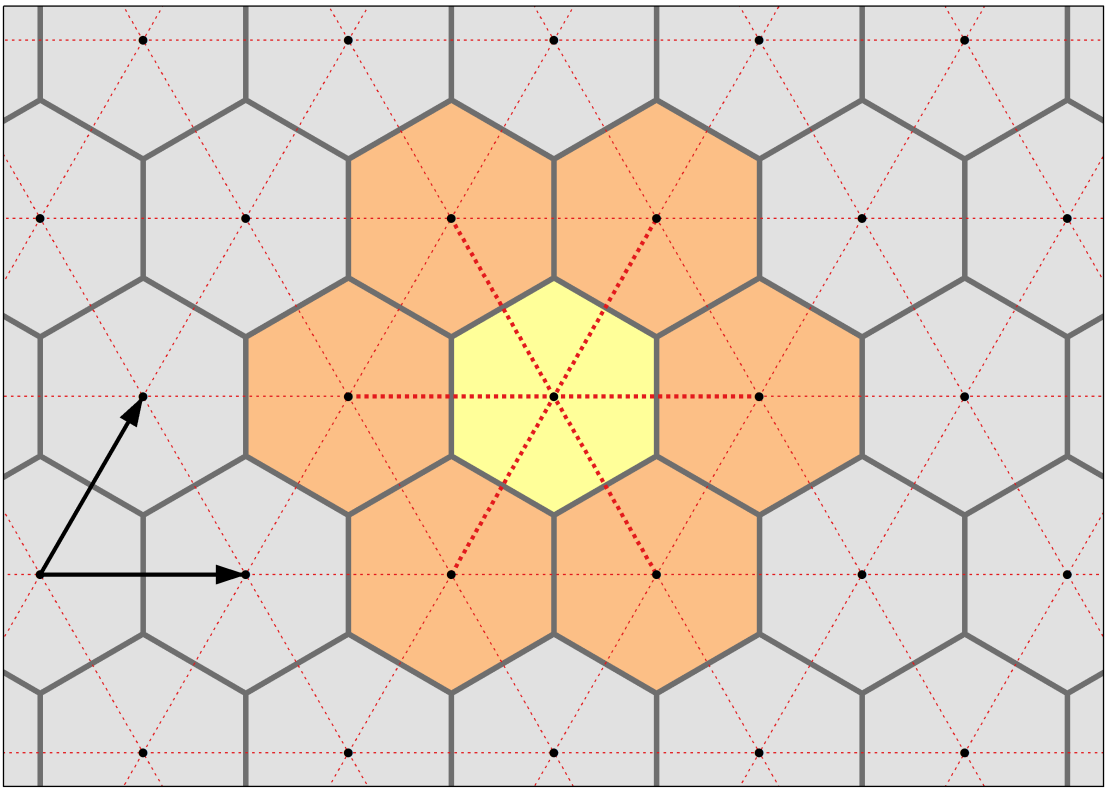


Figure 3.6.: The geometry of regular hexagonal tessellation. The black dots are the tile generators. The grey lines are the perimeter of the hexagonal cells. The black arrows represent the basis vectors of $\mathbf{R}_{hex}$. The neighbourhood of the yellow cell is coloured orange. All neighbours are adjacent across lines. The dual network is shown in dashed lines. The network lines associated with the yellow cell are shown thicker. All network lines are the same length. The network lines form the skeleton of a triangular array.

equally distant from the central cell. The edge only adjacency removes the connectivity problem. Adjacent neighbours of a given cell are separated by lines that are covered by a line radiating from the centre of the central cell. Hexagonal Voronoi diagrams are not degenerate.

Hexagons can be decomposed into six equilateral triangles or aggregated into rosettes that approximate hexagons. If the aggregation process is repeated, the perimeter of the aggregation becomes increasingly fractal as the number of tiles in the aggregation increases. Voronoi tessellations formed from the centroids of the rosettes are true hexagons with their orientation rotated (Middleton and Sivaswamy, 2005a).

The projection of a hexagon onto a line in direction θ can also be determined using Equation 3.7 where h is instead the distance between any two line adjacent neighbours. The similarity with triangular cross sections is due to the decomposition of hexagons into triangles. The cross section of a hexagon varies from 1 to $2/\sqrt{3}$ multiplied by the cell size.

Hexagonal tessellations are not as commonly used as regular rectangular tessellations or irregular triangular networks, partly due to the ubiquity of rectangular display and sensor hardware. Hexagonal sensors are available, although uncommon (Condat et al., 2008). Snyder et al. (1999) describe a hexagonal coordinate system based on tilted axes (u, v) and conversion to Cartesian (see also Subsection 3.8.3). Notwithstanding the predominance of rectangular and triangular tessellation, numerous advantages of hexagonal geometries are described in research papers; the remainder of this section details some of these.

The local neighbourhood isotropy facilitates neighbourhood operations (de Sousa et al., 2006), and the implementation of circularly symmetric kernels (Snyder et al., 1999). Hexagons are the closest of the regular tessellations to a circle. Therefore, the length of the cross section changes least with direction of transit. This is discussed with reference to flow direction in Subsection 8.3.1. Voronoi cells of uniform randomly generated points tend toward hexagonal geometry because they have on average six sides.

Hexagonal sampling generates finer spatial resolution than rectangular sampling for a given number of samples (Snyder et al., 1999; de Sousa et al., 2006), and the unit hexagon requires the least length of sides of the alternatives (Brimkov and Barneva, 2001). Hexagon are highly economical at packing space; if a collection of tangent circles with flexible boundaries are subjected to pressure a hexagonal grid is the lowest energy solution. There are many naturally occurring structures that utilise hexagonal packing including beehives and the human retina (Snyder et al., 1999). The Nobel Prize in Physiology or Medicine, 2014, was awarded to John O'Keefe, May-Britt Moser, and Edvard I. Moser for research that showed that mammalian brain cells use a hexagonal map of space to sense location (Nobel Media AB, 2014).

Brimkov and Barneva (2001) demonstrate that the discretisation of a line into hexagonal tiles is thinner than rectangular or triangular discretisations provided that the discrete line is tunnel free, i.e. the line can be traversed from end to end via only edge adjacent neighbours. Hexagons are good for pattern recognition in aerial photos or other objects without predetermined preferred directionality (Golay, 1969). Finally, flow directions are better preserved when up-scaling a hexagonal array compared to up-scaling a rectangular DEM (de Sousa et al., 2006).

3.6. Interpolation, resampling, conversion, and generalisation

In addition to geometry and structure, the third aspect of a surface model is the method of interpolation of values away from the point set (Subsection 3.6.1). Interpolation can be

used for resampling, for instance, converting between meshes with different geometry or to convert between different resolutions of the same mesh (Subsection 3.6.2). Interpolation can also be used to convert between different structures, for instance, converting from a TIN to a regular tessellation (Subsection 3.6.3). Subsection 3.6.4 discusses generalisation and distinguishes it from resampling.

3.6.1. Interpolation

2.5-D Surface models frequently provide estimates for values at locations outside the point set by interpolating values using appropriate functions of (x, y). Given a set of values at sample points, interpolation is the process of producing estimates for values at points not included in the original set.

Functional surfaces for interpolation can be generated from point sets, network graphs or tessellations. Rarely, a global function may be applied to the entire domain, for instance Yue et al. (2007) present a global surface interpolation method based on the fundamental Theorem of Surfaces. However, more usually, the interpolated surface will be formed piecewise from local functions. Whether global or piecewise, functional surfaces that are based on fitting equations to sample points generally do not follow the points exactly; they are smoother than the underlying data. Alternately, some functional surface models respect all points. Surface models that respect all points are useful for updating, inserting, and deleting points because the changes only have local effect (Gold, 2009). A large variety of equations may be used for interpolation including polynomial (local and global), bilinear, nearest neighbour, Inverse Distance Weighted (IDW), splines and regularized spline with tension (RST) (Chaplot et al., 2006). Interpolations that preserve shape and consider drainage structure, such as those used by the ANUDEM gridding algorithm discussed below in Subsection 3.6.3, produce surfaces that are more suitable for hydrological modelling than those that consider only local elevation.

A piecewise interpolation may be subdivided by the edges of a network or tessellation. Tessellations are effectively nearest neighbour interpolations of point sets. However, tessellations are not truly functional unless you ignore the edges and vertices or apply a rule to produce a single elevation value in these places. A continuous surface can be conveniently created from a conforming TIN by fitting a plane to the three vertices of each facet. Triangular conforming meshes avoid discontinuities at facet borders. Fitting planes to TINs is a bilinear interpolation; nearest neighbour, bi-cubic and bi-quintic interpolations are also used (Watson, 1992).

Interpolations may be developed using techniques from geostatistics, such as, point kriging, ordinary kriging, and universal kriging (Mitas and Mitasova, 1999). Geostatisticical

methods propagate errors in source data and provide information about the probability of the fit between the model and the sample points and can be used to create equal probability simulations (Hengl et al., 2008). Functional surface models can also be formed using functions that are defined in frequency space, such as, wavelets or methods related to the Discrete Fourrier Transform (Gallant and Hutchinson, 1997). Frequency space surfaces are often used to compress data and lend themselves to multi-scale analysis (see Section §3.8). Huang and Chen (2009) describe a fractal DEM interpolation method.

3.6.2. Resampling

Resampling produces a "target" point set from a "source" point set. The target and source point sets may differ in geometry and density but are both approximations of the same underlying surface. Resampling between a source point set and a target point set can be accomplished using a generator function that estimates the unknown underlying function by means of its samples in the source point set. The generator function is then sampled on the target point set (Condat et al., 2008). On 2-D Cartesian point sets, generator functions are typically obtained as tensor-product extensions of 1-D versions (Condat et al., 2008).

On non-Cartesian point sets, such as the hexagonal one, intrinsically 2-D generator functions are deployed: hex-splines (Van De Ville et al., 2004) or box-splines (Condat and Van De Ville, 2008). Middleton and Sivaswamy (2005a) summarises methods for resampling between hexagonal and rectangular arrays, including least squares spline resampling presented in Van De Ville et al. (2002). Faille and Petrou (2010) use splines to reconstruct images on hexagonal grids from samples. These methods require computationally expensive pre-filtering to produce interpolations more advanced than linear (Condat et al., 2008).

Generator function methods cause image degradation and are not reversible. They are not reversible because the generator function is intrinsically linked to the source point set only. Condat et al. (2008) present an alternative approach for regular arrays where source and target arrays are the same density, which seeks to achieve "Symmetric reversibility". The Condat et al. (2008) method consists of a maximum of three shear operations (see Figure 3.7) combined with a 1-D fractional delay operation to interpolate the values. In addition to converting between Cartesian and hexagonal arrays, it can convert between any two regular arrays, including rotations (Condat et al., 2008). The Condat et al. (2008) method is implemented to convert between rectangular and hexagonal arrays in Part III.

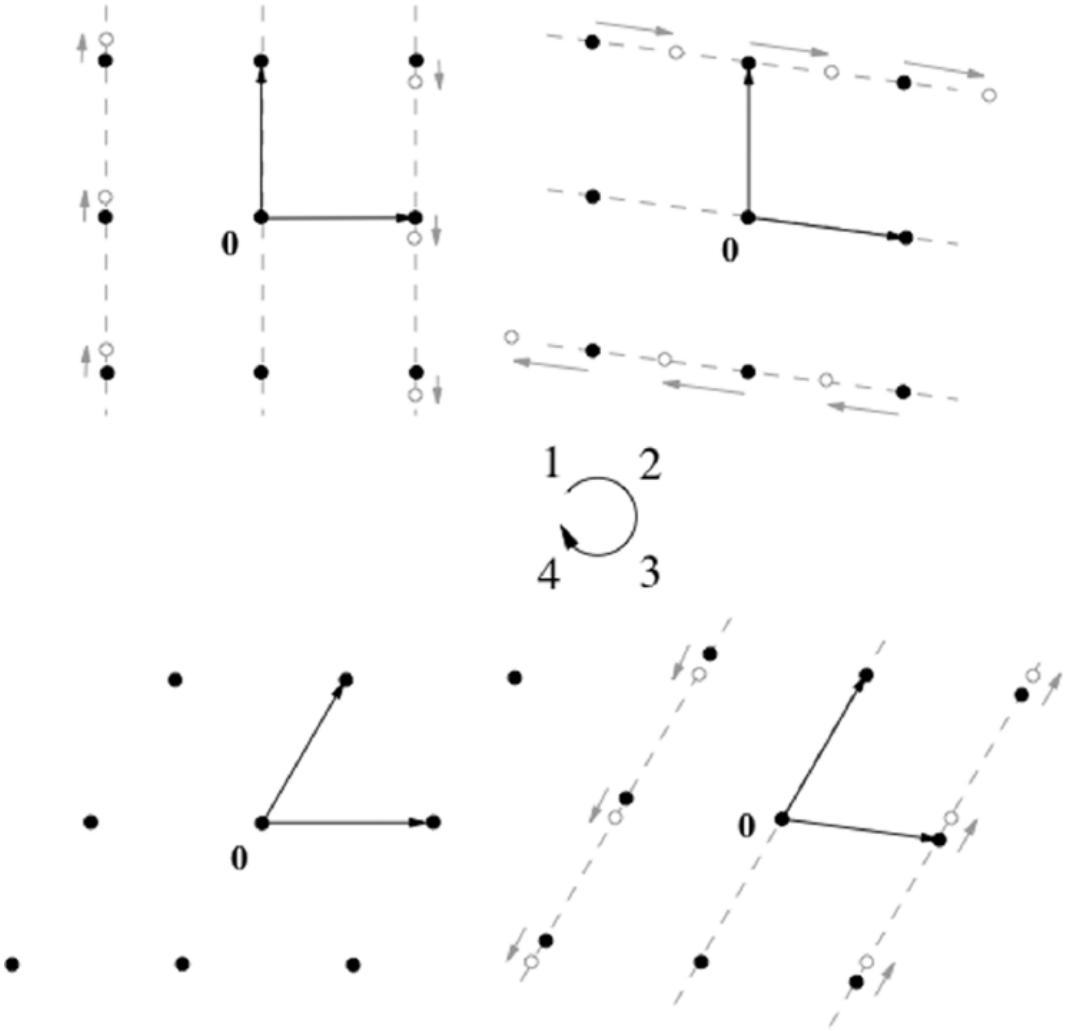


Figure 3.7.: Shear operations required to resample from rectangular to hexagonal geometry (from Condat et al. (2008)).

3.6.3. Structure Conversion

Gridding

Converting a point set into a tessellation can be as simple as determining the Voronoi diagram. Although, if a regular grid is sought it may be necessary to resample the geometry using the techniques listed in Subsection 3.6.1. There are a variety of techniques used to generate regular rectangular DEMs from source data, which are typically vector data consisting of points, break lines, and contour lines. Here, a vector to raster interpolation is called a gridding procedure. Methods for interpolating raster data from point or TIN are now being developed to use streaming techniques so very large data sets that do not fit in main memory of commodity computers can be processed (Isenburg et al., 2006; Agarwal et al., 2006).

Gridding procedures developed specifically for modelling terrain often include constraints that cause the generator function to conform to land forms. However, such methods are not necessarily suited for areas where the terrain is modified by human activity. Niemann et al. (2003) introduced a physically based model that simulates topography in dynamic equilibrium including the effects of tectonic uplift, fluvial incision, and hill slope diffusion. Agreement between the simulated topography and the elevation data is improved by iteratively adjusting a spatially varying erodibility parameter. Mass Balance (Grimaldi et al., 2005) extends the steady state physically based erosion model so that it produces slopes with variable curvature and 2nd moment self similarity, thus generating realistic profiles from sparse data.

Hutchinson (1989) introduced ANUDEM, which is a very popular algorithm for creating hydrologically enforced grid DEMs from sparse point and break line data. ANUDEM is a locally adaptive elevation gridding procedure that involves an iterative finite difference interpolation technique and a hydrological enforcement algorithm over successively finer grid divisions until the final resolution is reached. The ANUDEM procedure incorporates a variety of roughness penalties composed of surface curvature and potential that can be tuned to different data sources. The strength of the ANUDEM procedure is that it has proven effective in determining surface drainage structure even from relatively sparse data points. Hutchinson (2000) suggests a method of determining optimum resolution based on the available data. Yang et al. (2007) compares ANUDEM and TIN interpolation and optimizes ANUDEM parameters.

Over the years the ANUDEM algorithm has been improved to incorporate a greater range of data including contour data and LiDAR data and the hydrological enforcement algorithm, which seeks to create a connected drainage structure, has been altered (Hutchinson et al., 2009). The emergence of LiDAR data (Subsection 3.2.2) has motivated changes to the ANUDEM algorithm because the original algorithm assumed sparse noise free data. However, LiDAR datasets are frequently as dense or possibly denser than the desired grid resolution and LiDAR data are known to contain significant noise and systematic biases (Hutchinson et al., 2009). MacMillan et al. (2003) discusses some of the challenges of making DEMs from LiDAR.

Techniques from the field of geostatistics can also be used to grid elevation data. However, geostatistics are used to form topographic surfaces less commonly than splines because the results of geostatistical techniques are sensitive to sampling density and local extreme values (Hengl et al., 2008). Regression kriging is presented as an option to splines in Hengl et al. (2008). This method permits the use of auxiliary predictors, such as river lines, in the interpolation.

Triangulation

Triangulation is an umbrella term for the creation of TINs from a set of points. Zhu et al. (2001) and Braun and Sambridge (1997) describe generation of TIN from contours. TINs are often generated from grid DEMs, in fact, grid DEMs can be treated as a TINs (Palacios-Vélez and Cuevas-Renaud, 1986; Agüero et al., 2003). However, GRID to TIN triangulation is typically preceded by culling of grid cell centres to produce a subset of points for triangulation. Point selections are constrained to preserve slopes within a set tolerance. Common methods for selecting points for triangulation are Drop Heuristic and Very Important Point. Vivoni et al. (2004) compared Drop Heuristic and Very Important Point and found Drop Heuristic performed better. Wang et al. (2001) found that iterative point selection and triangulation is more effective than one pass methods.

Generally selected points are triangulated using Delaunay Triangulation (see Subsection 3.4.2). Delaunay Triangulation only considers the horizontal position; alternative data dependent triangulations such as Lawson's local optimization procedure also consider the z-value. Two non-Delaunay alternatives are constrained Delaunay Triangulations, which introduce linear features to constrain the edges of some triangles; and Greedy Triangulation, which minimises edge length.

Abdelguerfi et al. (1998) analyse triangulation and identify a good triangulation as one that represents the surface accurately, retains edge to edge connectivity; avoids slivers, which occur when the triangulation produces thin and slivery triangles; and avoid streaks, which appear when there are too many triangles at a given vertex. A variant of Scarlatos and Pavlidis (1992)'s algorithm shows the best overall performance (Abdelguerfi et al., 1998). However, Wang et al. (2001) compare existing methods of triangulation and find Delaunay to be the best.

3.6.4. Generalisation

Generalisation, in the field of spatial science, is a broad concept that encompasses various techniques that convert dense data into a more sparse form. One example of generalisation is up-scaling a raster dataset to a coarser resolution to form a multi-scale pyramid (Subsection 3.8.1). Alternatively, generalisation is used to describe both the cartographic techniques whereby vector data are simplified by removing nodes to represent objects with appropriate complexity for a given scale, and the substitution of data with a symbol for the same purpose. Due to the limited spatial accuracy of measurement and the finite but greater than point area required to measure distributed properties, all geographic data are generalisations (Zhang et al., 2014).

The conversion from a dense DEM to a sparse format, such as the conversion from grid Raster to TIN (Subsection 3.6.3), is also a form of generalisation. Generalisation is also required to integrate a fine dataset with coarse data from the same domain, such as adding LiDAR data to existing DEM models (Dalyot et al., 2008). Various methods that have been developed for simplifying polygonal surfaces (such as TINs) are summarised in Heckbert and Garland (1997). A summary of adaptive thinning techniques is given in Demaret et al. (2005).

Generalisation is not synonymous with resampling from a dense source point set to a sparse target point set because generalisation addresses scale. Even if the target point set of a resampling is less dense than the source, resampling will generate only point values. In contrast, a generalisation seeks to aggregate or integrate the values so that the new values are representative of a larger region than the original ones. Ai and Li (2009) assert that DEM generalization should be governed by geographic significance not geometric properties. Their proposed method preserves major valley structure while filling minor ones. Therefore, their method leaves resolution the same but removes complexity from terrain.

Generalizing can be applied selectively to different wavelengths within the terrain. For instance, Jenny and Hurni (2010) use Laplacian pyramids to interactively control generalisation levels at multiple scales using a graphic equalizer interface for 3-D perspective visualization. Döll and Lehner (2002) describe up-scaling methods for grid based outputs of hydrological geomorphometry, including drainage vectors. de Sousa et al. (2006) up-scale drainage vectors on a hexagonal grid.

3.7. Data models

Regardless of the structure and geometry employed to describe a surface, a data model must be employed to store, query, and update the surface description within a computer's memory. There are a great many ways to store digital spatial data, with new methods being developed all the time. A comprehensive survey is beyond the scope of this thesis. Instead, three broad categories will be described in a general sense: vector, raster, and tree structures.

3.7.1. What is a data model?

Digital data

Computer memory consists of strings of binary information: 0s and 1s; that encode standard data types, such as text, and numerical values. Binary representations of numerical values, such as floats and integers, have limited size, precision, and in the case of floats, accuracy. Memory must be allocated according to the type of data stored, therefore, storing 1 as a long integer requires the same amount of memory as storing 1000. Data structures are built to access these bits to produce data that is useful to people. Data models need to facilitate the storage, deletion, alteration, and querying of data to meet the need of users. Spatial data models are designed to minimise the computational expense of the simplest and most common spatial operations e.g. overlay and adjacency, whilst also providing versatility. There is an inevitable trade off between versatility and optimisation for a given application.

To be representative of a geographical area, a spatial reference is required to relate the (x, y, z) positions of the model to locations in geographical space. This is usually accomplished by equating (y, x) in computational space to Northing and Easting in a projected coordinate system or Latitude and Longitude in a geographic coordinate system. The coordinates are then related to a position on an ellipsoid via a projection. The ellipsoid is accurately fixed at some locations on the Earth. Alternatively, positions may be determined in three dimensions based on an Earth Centred, Earth Fixed reference frame. Typically, a spatial reference indicates the projection used and the precision with which coordinates are stored.

Conceptually, GIS data are generally in the form of "fields" or "objects" (Gold, 2009). Object data consists of unconnected entities such as hill tops or rivers as opposed to field data, which describes spatially continuous properties such as elevation or rainfall. Object data are typically stored in vector form and field data in raster form. DEMs are no exception and typically conform to these two primary data models of GIS.

Data models are related to but distinct from the structures and geometries of surface models. Some data models are closely associated with certain structure/geometry combinations, for instance, raster data model and square grid tessellations. However, they are not the same thing. It may be inefficient, but it is possible to store a square grid tessellation in a vector format.

The data model of a representation of a DEM can also differ from the underlying model. For instance, a computer screen visualises a vector contour layer as a raster image. Various styles of diagrams, tables, profiles and maps can display information derived from DEMs.

DEMs are typically displayed as 2-D maps in colour or grey tone. However, 1-D profiles and 3-D views are also common. Histograms can be used to show the frequency distribution of terrain measures, semi-variograms and spectral functions can be used to show scale dependence, output tables from correlation and factor analysis reveal redundancy among measures (Pike and Kimberly, 2005).

Implicit and explicit data models

Gerstner (2003) identifies four classes that can be used to describe the data structures of DEMs based on whether the coordinates and interpolation of the data models are implicit or explicit. Gerstner (2003)'s division is used below to classify the structure/geometry combinations discussed in this chapter. The nomenclature is slightly adapted to follow this thesis. A data model is implicit if the additional storage required is of an order $< O(N)$, explicit otherwise. Implicit data models are less flexible but have have lower overheads for storage and analysis. Global transformations do not change the class (Gerstner, 2003).

1. Coordinate and interpolation implicit: regular point sets and meshes.
2. Coordinate implicit, interpolation explicit: Quad trees, adaptive tensor product grids.
3. Coordinates explicit, interpolation implicit: TINs, Delaunay Triangulations, Semi regular networks.
4. Coordinates explicit, interpolation explicit. Data dependent triangulations, ad hoc tessellations.

3.7.2. Raster

The raster data model is analogous to writing words on a page (Goodchild and Mark, 1987). The term raster comes from the German raster, which means screen. The term implies that the data are stored as a series of rows, similar to the set of parallel lines that make up a television picture. Each row has the same number of entries so that the series of rows forms an array of rows and columns. The geographical location of the data value is encoded implicitly by the index of the value within the array, provided the position of at least one cell in the raster is known. The single known position can be combined with cell size in both the x and y dimension to determine the location of all tiles in the projected geographic space. Rarely, the array may also be rotated or otherwise systematically transformed between the computational and projected geographic space.

The term raster is usually used to denote a regular tessellation stored as a 2-D array, with rows being the first and columns being the second dimension. The data are in fact a linear sequence of values of the same type. However, because every row has the same number of columns it is very easy to traverse the second dimension. The number of columns in the array is the "stride" required to visit values in a given column. For instance, lets say we have a 2-D array with 5 rows and 10 columns. The stride is 10, therefore the values of one column are found by taking every 10th value.

Rasters are sometimes known as images. The most common coordinate system for raster images is (row, col), which corresponds to (-N,E) with the origin at the top left hand corner. Multi-band images may be formed by combining several raster layers representing different bands. For instance true colour images contain three raster layers.

Raster datasets are typically used to represent fields. Rasters can represent both continuous or discrete data. Rasters cannot efficiently represent sparse data because a value is required for every cell. A number is designated to represent No Data, often the highest or lowest value for the numeric data type used. Raster datasets are always rectangular. If the study area is not rectangular the remaining cells are filled with the No Data value. The inclusion of grid cells that represent non-study areas requires added computer memory and increased computational time. This problem is further compounded if a fine grid cell size is used to accurately represent landscape features (Jin and Wu, 1997).

The traditional raster format scans in row order horizontally from left to right across the page like reading in English. However, there are other space filling curves that linearise spatial data differently. These methods are motivated by a desire to group data together in computational space that is near in geographic space. One can reorder the row by row order of the raster into a fractal curve, which is still linear. Some space filling curves that are alternatives to the row order described above are: row prime order, N (morton) order, or Pi (Hilbert-Peano) order (Goodchild and Mark, 1987). Space filling curves are often related to hierarchical structures and can be the basis for a recursive subdivision of space (Goodchild and Mark, 1987). Raster data are often compressed to save disk space by various methods. Run order encoding can be applied to any space filling curve to save storage space.

3.7.3. Vector

The vector data model is analogous to drawing a map with a pen (Goodchild and Mark, 1987). Vector models are typically used to represent discrete objects. Data stored using the vector model encodes discrete objects with Cartesian (or other) coordinates in 2, or 3

dimensional reference frames to form points, lines, polygons, or polyhedra. Point clouds and contours are typically stored as vector data sets.

In a GIS, vector geometry data are often associated with attribute information by including the vector geometry as a geometry field in a relational database. Vector data can represent elevation either within the geometry field by providing a z-value to the vertices, or by attaching elevation values to each record as an attribute. Vector datasets may include several points, lines, or polygons in a single record to create multi-point, multi-line, and multi-part polygons respectively. Vector objects are frequently stored in a format that includes topological information to facilitate answering various spatial queries. Topology may be as simple as a way of storing which points are connected to form geometric networks or which polygons are adjacent.

Vector data has effectively arbitrary precision; although, it is practically limited by the precision of the spatial reference. When you consider the resolution of the coordinates used to define the nodes of the geometries, the vector model is comparable to an ad-hoc network graph with the majority of the nodes left unconnected and unattributed. However, attributes are associated to whole connected graphs not individual nodes.

Some surface models such as TINs do not neatly fit vector or raster formats. TINs require explicit encoding of position for each node, like the vector model, but are more suitable for representing fields, like the raster model. Representations of field data with irregular geometry, such as irregular TINs or Voronoi tessellations, are not compatible with raster data formats; the coordinates of the vertices need to be stored. Voronoi diagrams and TINs are usually structured as indexed arrays such as the winged-edge data structure. The identity of the neighbours is very important so data formats require explicit storage of topological information. The Quad-Edge data structure (Guibas and Stolfi, 1985) represents both the Voronoi and the dual at the same time.

3.7.4. Tree structures

An alternative to the raster data model and other space filling curves is the tree data model. Tree data structures use pointers to relate information in a hierarchical fashion. Tree graph structures, consist of nodes, branches, and leaves. Tree structures can be combined with space filling curves so that the branches on each node represent a hierarchical subdivision of space. Tree structures allow position variant resolution for data types that would otherwise be regular and are also conducive to multi-scale analysis (see Section §3.8).

Grid DEMs can be stored hierarchically using Quad-trees or N-trees (Goodchild and

Mark, 1987). Quad-trees divide space recursively until a desired level of consistency is reached. Therefore they support variable resolution. Quad-trees can be searched more effectively than rasters and generally require less storage space. The Quad-tree address is in Base 4. In a Linear Quad-tree, base 4 addresses are stored at leaf level and directly accessed. In a pointer-based quad-tree, four descendent nodes are connected by pointers to each parent, each branch of the tree represents a single digit of the quad-tree index. A binary representation of a Quad-tree index is a Morton address. Morton addresses can be converted to Cartesian coordinates by efficient bit operations. Conversion from Morton to Cartesian is accomplished by undoing bit interleaving (y first, then x) to get the Cartesian index, see also quadkeys used to index tiles in Bing maps (Microsoft Corporation, 2013).

In addition to Quad-trees, some other memory efficient hierarchical trees that can be used to represent surfaces are Point Quad-trees, Bi-trees, Hilbert Quad-tree, Four-Two tree, Sierpinski Bin-tree (Valle and Ortiz, 2011), Multi-resolution pyramid, Tiled pyramid (Platings and Day, 2004), and Gaussian and Laplacian pyramids (Jenny and Hurni, 2010).

Hexagons can be stored in a tree structure using recursive subdivision of hexagon like fractals (Lundmark et al., 1999). A hexagon rosette based tree data structure can be indexed using Generalised Balanced Ternary (GBT) spatial addressing system (Gibson and Lucas, 1982), or the HIP indexing system (Middleton and Sivaswamy, 2001). Alternatively, a hexagon point set can form the basis of an HoR quad-tree hierarchy (Bell et al., 1989). Recursive tree structures that form regular hexagonal, rectangular, and triangular arrays are described in Chapter 6 and form part of the datamodel of the RHSM.

3.8. Multi-scale models

A multi-scale surface model provides multiple representations of a surface at different scales. It is very likely that a given parameter at a given location will vary at different scales due to the effect of aggregation and the fractal nature of geographic phenomena (see Section §2.3). A common application of multi-scale surface models is to enhance visualisation of surfaces by varying the resolution depending on the location of the viewer relative to the surface model. A multi-scale hierarchical surface model can be used to generate realisations that meet a specified degree of precision. This section describes the structures, geometries and data model of hierarchical surfaces. Hierarchical adaptive methods for interpolation and surface modelling are summarised in Gerstner (1999).

3.8.1. Hierarchical structures

There are three basic types of multi-resolution DEMs, pyramidal, incremental, and graph structured (Gerstner, 2003). Hierarchical structures can also be formed in frequency space.

Pyramid models

Pyramidal models consist of a number of DEMs with different resolutions. The different levels are effectively different datasets that share a common spatial location, which can be layered to form a pyramid. The simplest such models are raster pyramids that have multiple representations of the same area at different cell sizes. Raster pyramids may be combined with data tiling to support very large data sets. The most common use for raster pyramids is to rapidly redraw raster data sets in an interactive map as the viewer "zooms" in to see fine details or "zooms" out to get a coarser overview. The values for raster pyramids are generated by summarising the cells in the finest resolution layer using statistical methods; such as, nearest neighbour, maximum value, minimum value, mean, or median. However, mathematical interpolation can also be used.

Incremental and graph structured models

Incremental models encode a series of individual steps to transform between the coarsest and finest representation; therefore, many more intermediate resolutions are possible than when using a pyramid structure. Graph structured models are formed by identifying and encoding the hierarchical dependencies of the individual steps in an incremental model. Graph structure allows local refinement, which means that the resolution can vary across the object (Gerstner, 2003). For instance, the resolution may become coarser with distance from the viewer. Such method are known as Level of Detail models (LOD). Usually, LOD models are used for real time or online applications and often for complicated objects. Therefore the data structures need to be very efficient.

LOD models can be built from irregular or regular point sets. Continuous approximation can be reconstructed from the vertices using radial basis functions, by making k-simplices (triangles in 2-D) with the points as vertices (de Floriani et al., 2005), or by visualising the points as splats, which are ellipses, hexagons, or rectangles (de Floriani et al., 2005). For an overview of LOD models, see de Floriani et al. (2005).

The process of extracting a realisation from an LOD consists of either top down refinement

or bottom up decimation to achieve a desired Level of Detail (LoD) (Gerstner, 2003).[3] The LoD may be uniform or variable in space (if graph structured). If the LoD is variable, the refinement is selective. Selective refinement applies the refinements required to achieve a user defined LoD in a user defined region of interest (Danovaro et al., 2006).

The LoD criterion can be represented as an equation with true or false outputs. For example, given the LoD criterion 'T' apply the minimum number of refinements required to satisfy T (de Floriani et al., 2005). Refinements may involve adding edges, nodes, and/or splitting and removing existing edges. Significant breakpoints represent new elevation data and are introduced to achieve the required accuracy. Insignificant breakpoints break an edge to maintain conformity but do not alter the elevation (Abdelguerfi et al., 1998). Regular meshes and semi-regular meshes are formed by subdividing a regular network or irregular network with the recursive application of the same basic refinement operator. Regular and semi-regular meshes are usually stored as trees or directed acyclic graphs (DAGs) (de Floriani et al., 2005).

Frequency space

Frequency space models are a very different approach. Bjørke and Nilsen (2003) demonstrate wavelet surface smoothing by altering the character of the fundamental wavelets. Wu and Amaratunga (2003) describe a multi-resolution wavelet TIN based on linear estimation and modified butterfly subdivision, intended for transmitting DEMs for web based services. Non manifold and variable dimension meshes are described in Danovaro et al. (2006). A generalised framework for integrating different resolutions based on fuzzy sets is described in Worboys (1998). This system allows modelling of imprecision due to finite resolution.

3.8.2. Hierarchical geometries

A hierarchical multi-scale surface model relates cells at different levels of detail. Usually, with a one to many relationship where one "parent" at a coarse level of resolution relates to many "children" at the finer level of resolution, thereby, forming a tree structure. The finest scale level, which has the most and smallest cells, may be called level 0, the atomic level, or reference mesh. The coarsest level may be called the base or root mesh (de Floriani et al., 2005). [4] The number of levels in a hierarchical tessellation is limited to

[3] Note the distinction between LODs, which are multi-resolution models and LoDs, which are realisations at a specific resolution.

[4] However, care is required as some authors call the finest mesh the base mesh. In Parts II and III the finest mesh is referred to as the base level.

the number of levels required for the coarsest level to be only a single tile and no further agglomeration is possible.

Sahr et al. (2003) define various characteristics of hierarchical tessellations including congruent, nested, aligned, aperture, and self similarity. If the hierarchy is congruent, the borders of a coarser level parent are covered by the borders of its finer level children. A hierarchy is nested if all the children are completely covered by the parent (Abdelguerfi et al., 1998). Generally, if a hierarchy is congruent it will be nested. If the hierarchy is aligned, the centroid of the parent is covered by the centroid of one of its children. The aperture of a hierarchy is the ratio of the size of the parent and the size of the child. Aperture is only applicable to regular tessellations because it assumes all tiles of a given LoD are the same size. Tessellation Parameter is a more general term used by Tsui and Brimicombe (1997) that indicates the number of child tiles in the parent. A hierarchy is self similar if the children and parents have similar shapes.

For both regular and irregular tessellations, many hierarchies are possible. Quad-trees have an aperture 4 agglomeration that forms congruent, non aligned, self similar hierarchies from regular, conforming, rectangular tessellations (Figure 3.8 1A). Another regular rectangular hierarchy is the non-tree, which is an aperture 9, congruent, aligned, self similar hierarchy (Figure 3.8 1B). Regular hexagonal tessellations can form aperture 7, congruent, aligned, non self similar hierarchies (Figure 3.8 2A) or aperture 7, non congruent, aligned, self similar hierarchies (Figure 3.8 2B) among others. Non congruent hierarchies can permit a wide range of apertures (Sahr et al., 2003). The HoR quadtree sub division consists of either four hexagons or four rhombi tessellations based on a hexagonal point set. HoR combines desirable geometric properties of both hexagons and rhombi: the uniform adjacency of hexagons, and the divisibility and uniform orientation between hierarchy levels of rhombi (Bell et al., 1989).

A recursive Voronoi method presented by Boots and Shiode (2003) and further developed in Feick and Boots (2005), involves using the elements of the tessellation as well as some of the original generators to create the next set of generators. Tsui and Brimicombe (1997) introduce Adaptive Recursive Tessellation (ART). ART decomposes space into an aligned rectangular hierarchy. However, the cell shapes and aperture varies between levels.

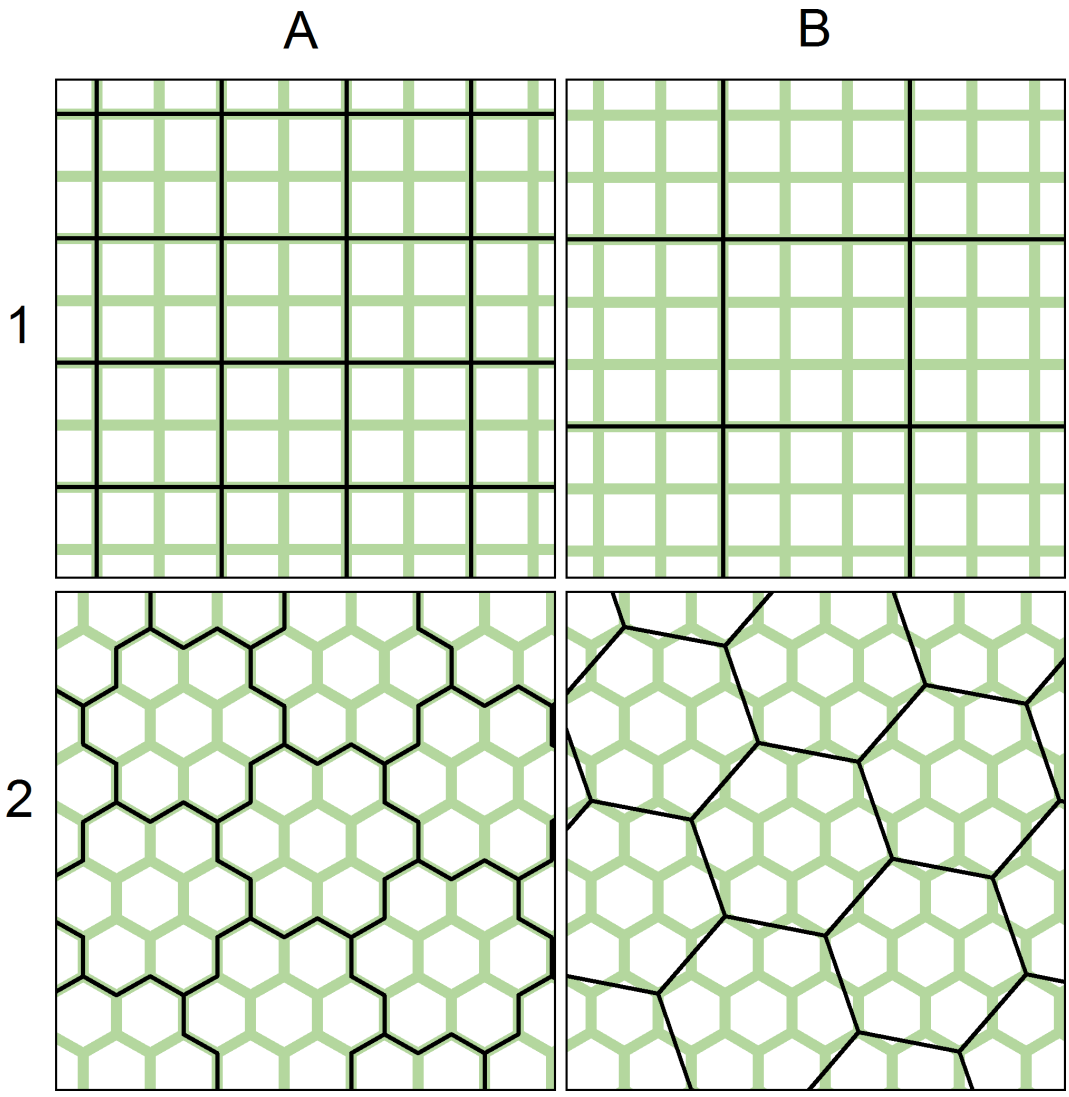


Figure 3.8.: Hierarchical geometries.1A is the quadtree structure, it is self similar, aperture 4, non-aligned, congruent and nested. 1B is the non-tree structure it is self similar, aperture 9, aligned, congruent and nested. 2A is non-self similar, aperture 7, aligned, congruent and nested. 2B is self similar, aperture 7, aligned, non-congruent and non-nested, however, it is semi-nested in the sense that the centroids of the children are invariably contained by the parent, although great-grandchildren may not be.

Hierarchical triangulation

Several common methods of refining triangular structures are listed below. In addition, the Scarlatos and Pavlidis (1992) algorithm uses five different strategies for dividing a triangle by mixing centre splits with edge splits. However, this introduces a variable tree structure as each triangle may be treated differently and is not conforming.

1. Right Angle Binary Tree Triangulation (RA-BTT) forms two right angle triangles by splitting one edge of the parent triangle and connecting it to the opposite corner.
2. Ternary Tree Triangulation (TTT) involves splitting the parent triangle into three parts by finding an internal point and connecting the original vertices to this point. TTT is conforming and has simple tree structure (always 1:3) but streaks and slivers are produced. Also if the internal point is too close to an existing edge, a sliver will be created.
3. Quaternary Tree Triangulation (QTT) involves splitting a triangle into four parts by finding the midpoints of each side and connecting them to form a central triangle and three edge triangles. Aperture four, self similar, regular triangular hierarchies are an example of QTT. QTT Makes near equilateral triangles, which are aesthetically pleasing with no streaks and slivers, and has a simple tree structure (always 1:4). However, all triangles need to be decomposed to the same level, otherwise it is not conforming.
4. $\sqrt{3}$ Subdivision divides triangles instead of edges. One subdivision generates a rotated non congruent mesh. A second subdivision generates a non aligned, congruent aperture 9 mesh (Kobbelt, 2000).
5. Delaunay Pyramid Triangulation is formed by adding another node to an existing triangulation and recalculating the Delaunay Triangulation de Floriani (1989). Delaunay Triangulations are useful for many purposes but as a hierarchy they do not satisfy the nesting property; therefore, they are not easy to store and the entire mesh may be affected by a single insertion. A constrained Delaunay can be used to enforce nesting.

Hierarchical triangulation can also be based on regular geometries, effectively combining quad-trees with TINs (Von Herzen and Barr, 1987). Pajarola (2002) presents a summary for terrain visualisation of tiled blocks and nested regular grids, quad-trees and triangle bin-trees triangulations, as well as cluster-based approaches. Regular meshes are more space efficient due to implicit encoding (de Floriani et al., 2005). Not withstanding the storage benefits of regular structures, retaining conformity in regular hierarchical triangulation generates many additional triangles and accuracy is restricted.

Global Hierarchical geometries

Hierarchical surface models can be referenced to an approximation of the entire earth, for instance, an ellipsoid or platonic solid (Sahr et al., 2003). Of the platonic solids, the icosahedron and dodecahedron are the most spherical. However, these polyhedra are not symmetrical and do not align with the poles and equator. The octahedron provides symmetry and has nodes and edges that align with the poles and equator. Sahr et al. (2003) define and describe a general process for choosing a Discrete Global Grid System based on five design choices, that lead to an appropriate structure. Global Hierarchical tessellations tend to have some extraordinary nodes, which cause a small number of tiles to have a different shape (Goodchild and Shiren, 1992; Sahr et al., 2003).

One example of global hierarchy is the Quaternary Triangular Mesh (QTM) global location coding model (Goodchild and Shiren, 1992). In the QTM the globe is projected onto the surface of an octahedron. Each face of the octahedron is an equilateral triangle, which is recursively subdivided into four smaller equilateral triangles by Quaternary Tree Triangulation (QTT). Other global level of detail models are presented in Platings and Day (2004). Hexagonal hierarchies can be applied globally including the polar regions using icosahedron-based Geodesic DGGS, with less areal distortion than rectangular systems (Sahr et al., 2003). The Open Geospatial Consortium has recently published a Discrete Global Grid System Standard (Purss et al., 2016).

3.8.3. Hierarchical data models

LOD Data models

LOD data models have benefited from considerable development within the field of computer visualisation. As discussed in Section §3.7, when designing data models, there is a trade-off between efficient storage and querying on one hand, and generality and flexibility on the other (de Floriani et al., 2005). Danovaro et al. (2006) identify three data model considerations:

1. generality and flexibility,
2. efficiency (query performance), and
3. compactness.

Other considerations are out of core models and client/server architectures (de Floriani et al., 2005).

The information that needs to be stored consists of coordinates, structure, and attributes. Multi-triangulation is a general framework for explicit irregular structures (Puppo, 1998;

Cignoni et al., 1997). Multi tessellation extends multi-triangulation into three or more dimensions (de Floriani et al., 2005). The basic model of an LOD data structure consists of a base mesh (coarse), a reference mesh (fine), a set of updates, and a dependency relation among the updates, usually stored as a directed acyclic graph (DAG) (de Floriani et al., 2005). Other attributes can be attached such as error measures, textures and normals (Danovaro et al., 2006).

A given update (u) consist of two sets of cells ($u+$ and $u-$) that are finer and coarser resolution representations of a given area. Explicit data structures encode the vertices of the updates in an indexed data structure, which requires large storage overheads. Implicit structures encode procedural updates required to form triangles. Implicit structures are more compact but can take longer to perform selective refinement (Danovaro et al., 2006).

Triangulation can be encoded edge-based or triangle-based. The most common data models are the indexed data structure and index data structure with adjacencies, including winged edge, doubly connected edge, and half edge structures (de Floriani et al., 2005). Oct-trees and binary partitions are the most common, usually built top down (de Floriani et al., 2005).

Platings and Day (2004)) describe a tiled quad tree (TQT) for level of detail rendering global terrains, which stores tiles on each branch rather than a single value. TQT uses external memory. The balance between using smaller and larger tiles in TQT is a balance between fewer large tiles that bring unnecessary data and more smaller tiles that create more reads. Gerstner (2003) describes binary bit operations to resolve common nodes in a triangular bit tree, and methods to compress multi resolution data and extract realisations without decompressing.

Linear hierarchical reference systems

DAG models are explicit systems that store coordinates. However, Implicit LOD models based on regular hierarchies can be created using linear referencing systems. For instance, several indexing systems have been described that exploit aperture seven hexagonal hierarchies to concisely represent position including the Hexagonal Image Processing (HIP) system described in Middleton and Sivaswamy (2001) and a system based on Generalised Balanced Ternary (GBT) (Gibson and Lucas, 1982). Both HIP and GBT referencing systems aggregate rosettes of seven hexagons into larger pseudo-hexagons. Another example is the aperture-4 rectangular hierarchy, known as Quadkey that is used in the Bing Maps Tile System (Microsoft Corporation, 2013). Quaternary encoding of linear hierarchical systems was developed by Gargantini (1982) as a data compression method.

All these systems describe two-dimensional location with a single ordinate, the length of which is proportional to the LoD. In addition, each position ordinate begins with the ordinate of the corresponding pixel of the coarser LoD that contains it (Wright et al., 2014). Such systems are implicit because the values can simply be stored in a linear sequence with the index of the value in the list being translated into position. The hierarchy is also implicit because omitting the last digit of an ordinate gives the ordinate of its parent. The HIP referencing system is described in more detail in Section §6.1.

This discussion of hierarchical data models concludes this review of DEMs. The next chapter will explore algorithms that derive parameters from surface models and reinforce the understanding of specific DEMs as examples from the range of possible structures, geometries, interpolations, and data models introduced in this chapter.

4. Hydrological geomorphometry

The previous chapter discussed DEMs: the models that store digital approximations of elevation surfaces. With the emergence and widespread application of technologies to capture and store DEMS, a set of procedures and methods for the quantitative analysis of DEMs has developed; this developing science is variously known as geomorphometry, topographic analysis, terrain analysis, or digital terrain analysis (Gallant and Hutchinson, 1997). Geomorphometry is aligned with the practice of surveying but updated with the prevalence of modern data collection, processing techniques, and equipment. For a brief history of terrain analysis, see Pike (2002). See Wood et al. (2009) for detail on geomorphometry software packages, such as Whitebox GAT (Lindsay, 2005), and TARDEM/TauDEM (Tarboton, 1997, 2008).

The outputs of geomorphometry can be classified as variables or objects. Variables are quantitative fields. Section §4.1 describes classification of variables and identifies and describes some geomorphometric variables that are relevant to hydrology. Objects are defined features in the landscape. Section §4.2 identifies and describes some geomorphometric objects that are relevant to hydrology.

Section §4.3 outlines a process of geomorphometric catchment analysis involving the variables flow direction (Subsection 4.3.1) and flow accumulation (Subsection 4.3.2), which are adapted for a hierarchical DEM in Chapter 8. Section §4.4 details hydrological conditioning, which is a collection of techniques that adjust surface models to improve the results of geomorphometric catchment analysis.

4.1. Geomorphometric variables

This section categorises geomorphometric variables[1] and describes some variables that are relevant to hydrology. Further variables related to catchment analysis are described in

[1]The terms variable and parameter are sometimes used interchangeably but it is useful to distinguish them in the context of hydrological models. Variables are representations of fields that vary during a model run and are not usually modified during model calibration. For instance, simulated factors in the model such as heads or fluxes; or external factors input as time series such as precipitation. Whereas, parameter can be defined as any factor that is not calculated by the model but is instead input prior

detail in Section §4.3. Variables are quantitative fields that are continuous over a given domain and are frequently modelled in GISc as raster layers. Variables can be determined from elevation data as fully distributed fields or as summary values of a larger area. It has been proposed that, given a lack of direct information, geomorphometric variables could be used as substitutes for other data types such as soil types (Shary et al., 2002) or land class (Franklin, 1987).

4.1.1. Categorisation of variables

Strahler (1952) introduced the distinction between gravity geomorphological processes and molecular ones. This division was refined by Shary et al. (2002)) to field specific variables that reference a field, usually gravity or solar radiation; and field invariant, which do not. Gravity on the surface of the Earth is a pervasive and constant force pulling liquids and solids toward a lower energy state, resisted by friction of the surface and subsurface; consequently, most of the variables relevant to hydrology are field specific. Field specific and field invariant variables represent the land surface or geometrical form respectively (Shary et al., 2002).

Shary et al. (2002) further categorised geomorphometric variables as belonging to three classes based on the area that must be considered to determine them. The area classes are local, regional, and global. The field classes are field specific and non-field specific. Therefore, there are six classes in all. Local variables are based on analysis of an area which was defined prior to determination; for instance, a 3×3 grid neighbourhood. Regional variables consider an area that is not defined until the variable is calculated, and global variables require consideration of the whole Earth.

4.1.2. Shape variables

Slope

Slope is the first derivative of elevation. Slope is field specific and is usually determined relative to gravity. Slope is important for Hydrological and Hydraulic modelling because it influences the velocity of surface water flow. The slope of a 2-D surface is direction dependent: it varies with direction of travel in the horizontal plane. Jones (1998) defines slope as the gradient, in the direction of a flow path, of a plane tangent to a surface. A flow

to the model's simulation and may be subject to alteration during calibration. Parameters are usually constant within a single run of a model but can vary with time (e.g. seasonal). Under this definition, the geomorphometric variables described here act as parameters in hydrological models. Therefore, geomorphometry is a method to deduce spatially variable parameters from measured elevation data.

path is a path of of maximum slope. Slope can be determined analytically on differentiable surfaces or estimated on other surfaces by a local analysis of a surface approximation from generated sample heights. Consequently, on non-differentiable surfaces, slope also varies with the sample spacing used to form the surface approximation.

Jones (1998) compared several methods for slope determination on grid DEMs and found that the method of Fleming and Hoffer (1979), which is given as an algorithm in Ritter (1987) performed best. Raaflaub and Collins (2006) undertook a Monte-Carlo analysis of the sensitivity of slope algorithms to elevation errors and found that algorithms that consider neighbourhoods with more members are less sensitive than those that consider fewer neighbours. Meyer et al. (2001) described slope calculation from irregularly spaced data points. Mizukoshi and Aniya (2002) presented a method for slope calculation from contours.

Curvature

Curvature is the second derivative of elevation. Shary et al. (2002) set out a system of twelve curvatures based on three primary curvatures, which are mean curvature, difference curvature, and unspherecity. Of the twelve curvatures, seven are field specific and five are field invariant.

4.1.3. Accumulation variables

Contributing area

Contributing area is a regional, field specific variable that represents the area that potentially contributes surface flow to a specified line or point. Contributing area is the magnitude of the area as opposed to the spatially defined region, which is a catchment object (see Subsection 4.2.3).

Contributing area is sometimes referred to as up slope area or upstream area and is analogous to flow accumulation, which is discussed in Subsection 4.3.2. Contributing area can be subdivided into Total Contributing Area (TCA), and Specific Contributing Area (a). TCA is the area of catchment contributing flow to a segment of contour line. Whereas a is the TCA at a point. Hutchinson and Dowling (1991) defined a as:

$$a = w \overset{\lim}{\rightarrow} 0 \frac{A}{w} \tag{4.1}$$

where A is the area of land surface vertically projected onto the horizontal plane, between two slope lines that originate at a common hilltop, bounded at the lower end by a contour segment of length w. For a section of contour, a can be approximated without the limit as:

$$a \approx \frac{A}{w} \tag{4.2}$$

In contrast to contributing area, the dispersal area or downstream area is the area that a point, line, or area flows into. Catchment and dispersal area estimates for hydrological modelling should also consider weighting areas that are more and less likely to produce run-of (Tarboton et al., 1991).

Topographic Wetness Index

Introduced by Beven and Kirkby (1979) the Topographic Wetness Index (TWI) is regional variable derived from slope and Specific Contributing Area (see Subsection 4.3.2). TWI is a measure of an area's propensity to become saturated and is also related to soil strength and surface run-off. TWI is calculated as:

$$\ln\left(\frac{a}{\tan p}\right) \tag{4.3}$$

where a is the upslope Specific Contributing Area per unit contour and $\tan p$ is local slope angle.

4.1.4. Fractal variables

Another class of geomorphometric variable that has attracted academic interest is related to fractal geometry. For a list of references on self-similar and fractal properties of streams and topography see Pike (2002). Some fractal geomorphometric variables are fractal dimension (D), Hurst Exponent (Cieplak et al., 1998), and Rank-size property (Goodchild and Mark, 1987). D can be estimated using various methods, including cell counting, variogram, dividers, and Fourier. There are some differences between D values determined with different methods.

Klinkenberg and Goodchild (1992) and Goodchild and Mark (1987) argue fractal properties should be used as spatial norms not verifiable models due to inconsistent fit between topographic data and fractal models; i.e. self similarity models are a good fit in some

areas and a poor fit in others (Klinkenberg and Goodchild, 1992). Poor fit of topographic data to fractal models could be due to poor estimation method or inappropriateness of the model (Klinkenberg and Goodchild, 1992).

4.2. Geomorphometric objects

Geomorphometric objects are discrete entities that can be determined by analysis of elevation surfaces. Objects are defined features in the landscape such as hills or rivers and are conveniently represented in GIS databases as vector objects. Objects can have both positional error and categorical error (Hagen, 2003). Examples of geomorphometric objects that are relevant to hydrology include pits, catchments, river networks, hills, and buildings. Deng (2007) gives an ontological classification of map objects. MacMillan et al. (2004) define a hierarchy of spatial entities.

There are many approaches to define land form objects from DEMs including both supervised and unsupervised classification. MacMillan et al. (2003) describes extraction of objects from high resolution DEMs. Unsupervised classification by vector agent is discussed in Borna et al. (2014). Dřaguţ et al. (2011) examine multi-scale analysis of variance to extract objects. Burrough et al. (2000) suggested that fuzzy techniques, for instance fuzzy-k means, are appropriate because land form classes overlap.

4.2.1. Pits

Pits, also known as sinks, are local minima in the surface. In tessellations, pits are cells with no lower neighbours. In the context of surface hydrology, pits are pixels water can flow into but not out of. Pits form discontinuities in drainage and affect hydrological models by trapping flows. Pits are typically removed from surface models prior to hydrological geomorphometric analysis (see Subsection 4.4.1). Pits are one of the critical features of land surfaces defined by Pike and Kimberly (2005); the others are peak, pass, pale (a high point between two pits), ridge, and course (channel). Not all pits are false; pits in urban areas may coincide with the presence of catchpits.

4.2.2. River networks

All catchments have a drainage network that concentrates the flow along various paths. River networks represent the part of a drainage network dominated by channel flow as

opposed to hill flow. It is often assumed that below ground networks roughly follow surface networks.

For a list of references regarding research on extracting drainage lines and catchments from DEM see (Pike, 2002). Olivera et al. (2000) and Renssen and Knoop (2000) describe examples of automated river routing extended to global scale. Liu and Snoeyink (2005) define river networks on a TIN as local channels and the trickle paths of discharges from saddle points. The self similarity exhibited by river bifurcation suggests that river networks should be suitable for analysis by fractal geometry. Cieplak et al. (1998) describe self similar and self affine fractal models of river networks.

Channel networks in natural catchments can be further characterised numerically using the Horton-Strahler system. The composition of the stream system of a drainage basin can be expressed quantitatively in terms of stream order, drainage density, bifurcation ratio, and stream-length ratio (Horton, 1945). Strahler (1952) refined the Horton order to avoid ambiguities creating the Horton-Strahler stream number. This principle was extended to other hydraulic properties by Leopold and Miller (1956), Shreve (1966) and others, see Tarboton et al. (1991) for more detail.

Algorithms to detect river networks are sometimes called channelisation. There are two main categories of river network algorithms: stream-head methods and cell classification.

Stream-head methods

Stream-head methods involve identifying stream-heads, which are the locations on a drainage network where channel flow becomes dominant, and then reclassifying all the downstream cells in the network as channel. Stream-heads can be viewed as scale dependent (Band, 1986; Quinn et al., 1995) but are frequently described as distinct geomorphological features (O'Callaghan and Mark, 1984). Stream-head methods require a connected network. Lin et al. (2008) discuss software that lets users define channel heads on screen.

A typical strategy for defining stream-heads is to use a constant threshold for contributing area (O'Callaghan and Mark, 1984; Garbrecht and Martz, 1993). Constant threshold techniques can be combined with minimum stream length to avoid short side chains. Maps are used to tune parameters in Garbrecht and Martz (1993) and Wang and Yin (1998), who adjust thresholds to calibrate stream length. This technique is of questionable utility if automated generation depends on the pre-existence of authoritative maps, unless the parameters are applicable to other areas.

Rather than using a constant stream threshold for the entire DEM, Vogt et al. (2003) use additional environmental characteristics from other data sources about climate, ve-

getation cover, terrain morphology, soils and lithology to create land classifications which have different stream threshold values applied. Tarboton et al. (1991) suggest identifying stream-heads as the smallest area where elevation related scaling laws, slope scaling and constant drop still apply. However, Gandolfi and Bischetti (1997) found poor correlation between slope and stream-head threshold.

Cell classification

An alternative technique to stream head identification on networks are cell classification methods, which date back to Peucker and Douglas (1975). Cell classification methods involve classifying all cells in the dataset as stream or non-stream based on some criteria and then connecting the stream cells to form a river network. Cell classification can be combined with thinning (Band, 1986; Dinesh, 2008) or Skeletonisation (Meisels et al., 1995) to produce stream networks.

Pirotti and Tarolli (2010) utilise land form curvature as an approach for channel network extraction from LiDAR data. Smoothing and window size need to be matched to feature size. Heine et al. (2004) determine whether a cell in an array is a stream based on a logistic regression of head area slope, mean slope, mean plan curvature, and mean profile curvature. Franklin (2010) suggests defining rivers by their higher dimensional structure; i.e. whether they are local minima of a V-shaped cross section. Chorowicz et al. (1992) add a profile scan technique to network flow routing methods to produce networks that include areal features based on local concavity.

4.2.3. Catchments

A catchment is the area covered by a connected drainage network defined by its outlet, which may be arbitrary, e.g. the confluence of channels, the low point of a basin from which water cannot escape, or the point at which a certain upstream area threshold is reached. Watershed is an alternative term for a catchment that is often used interchangeably. A catchment defined by a pit is called a basin (Liu and Snoeyink, 2005). The outlet of a catchment or basin is also known as a pour point. The lowest point on the perimeter of a basin is called the pour point of the basin. A basin's pour point is the point where pits will overflow and therefore is used for breaching (see Subsection 4.4.1).

The determination of catchment boundaries was traditionally done manually by interpreting surface contours on an appropriately scaled topographic plan. Modern techniques typically involve automated processing of a Digital Elevation Model (DEM). Excessive quantities of small catchments in a raster image can be considered over segmentation

(Moga and Gabbouj, 1998) caused by false minima (pits). Algorithms to determine catchments can be used to define hills by multiplying elevations by -1 before running the algorithm (Shary et al., 2002).

Catchments are typically defined from flow direction arrays. Any technique that creates an all inclusive connected drainage network (such as the Single Restricted Flow Direction algorithms described in Subsection 4.3.1) can conveniently be used to determine the catchment. In a connected flow network, any point on a surface that is not a catchment boundary or a pit is part of the drainage network and all points on the drainage network flow to exactly one pit or pour point.

Some specific implementations include the one pass method (Fairfield and Leymarie, 1991) and flow climbing recursion (Lin et al., 2008). Magalhàes et al. (2012) employ an efficient Queue based method to determine catchments and flow accumulation concurrently. Such methods are easily extendible to to other regular tessellations, Voronoi diagrams, and network graphs.

Techniques that rely on the identification of a continuous downhill path are collectively called steepest path techniques. However, flow paths cannot usually be defined mathematically in a discrete digital environment (Vincent and Soille, 1991). Digital discrete algorithms rely on their definition and should seek to closely match analogue solutions. Wright and Leonard (2012) found that multiple applications of the stochastic Rho8 flow routing model (see Subsection 4.3.1) can provide important information on the uncertainty of catchment boundary locations caused by modelling errors. An alternative technique models catchment boundaries as the result of inundation. These techniques may be called immersion techniques (Couprie et al., 2005; Roerdink and Meijster, 2000; Vincent and Soille, 1991).

A triangulated catchment algorithm is presented in McAllister (1999). An example of hierarchical catchments from TINs is presented in Liu and Snoeyink (2005). A contour catchment algorithm is presented in Menduni and Riboni (2000).

Getirana et al. (2009)) describe statistics for comparing catchments: added area, subtracted area, total mismatched area, and relative area (%). Wright and Leonard (2012) described a catchment comparison method developed from a method to quantify contour differences in Reinoso (2010).

4.2.4. Buildings

Automated delineation of buildings from elevation data is useful for hydrological modelling. Weidner and Förstner (1995) discuss extracting buildings from DEMs. Borna et al.

(2014) discuss a vector agent approach that combines both imagery and elevation data. Tse et al. (2007) discuss extracting buildings from Voronoi/TIN data structures.

4.3. Catchment analysis

Geomorphometry has developed a collection of tools and work flows that process elevation data to produce hydrological variables and objects. A typical example of geomorphometric hydrological analysis may involve the following steps[2]:

1. Hard hydrological conditioning, which alters elevation values to remove sinks and/or incorporate ancillary data (Section §4.4).
2. Flow direction, which determines the expected direction water would flow on the surface under the influence of gravity at a specific location (Subsection 4.3.1).
3. Soft hydrological conditioning, which alters flow direction or other derived properties of the DEM to remove sinks and/or incorporate ancillary data (Section §4.4).
4. Flow accumulation, which determines the magnitude of the area that could potentially contribute flow to a given region, point, or line (Subsection 4.3.2).
5. Catchment, which is a nominal classification that identifies the sink or outlet to which a region potentially contributes flow (Subsection 4.2.3).

Various algorithms have been published in the literature to perform this analysis. Some algorithms combine two or more of the above steps into a single process. For instance, RWflood (Magalhàes et al., 2012) combines sink filling, direction, and accumulation in a single process. Further steps may be added to the five step process or some steps omitted, particularly hydrological conditioning. Deterministic flow direction algorithms provide precise, repeatable catchment boundary definitions for a given DEM. However, despite this level of precision, various data and modelling uncertainties can affect the accuracy of the result (Wright and Leonard, 2012). For summaries and assessments of geomorphometric hydrological flow routing, see Wilson et al. (2008) and Erskine et al. (2006).

The five step hydrological analysis produces two variable fields: flow direction and flow accumulation; and two object sets: pit locations and catchments. These fields and objects can be further processed to determine catchment boundaries, channel networks, topographic wetness index, and more. These parameters are important inputs to distributed hydrological models (Gruber and Peckham, 2009). The computed flow direction, flow

[2]In the following discussion, the catchment analysis steps are ordered thematically; not in the order they are listed here, which is the approximate order of execution.

accumulation, and catchment rasters provide the spatial location and extent of a given catchment, and identify those pixels to which the majority of other pixels "flow into". These pixels represent situations where saturation and channel formation are most likely to occur (Wright and Leonard, 2012).

The above process is a static analysis, which can be extended to time dependent modelling using decay modelling or the de Saint Venant equations (Section §2.2). Geomorphometric analysis does not typically consider momentum or pressure and, therefore, is commensurate with the kinematic simplification of the de Saint Venant equations (Kampf and Burges, 2007). The process can also be extended to include randomness.

Catchment analysis algorithms are often presented as complete packages and compared on this basis. Frequently, however, they can be broken into the steps above, which can be assessed separately. Breaking algorithms into stages can be beneficial to better assess what parts of an algorithm work well and to combine effective parts of different algorithms. By making such a separation the range of possible techniques increases.

A systematic understanding of surface models helps to categorise algorithms by the surface model they are designed for: point, tessellation, or network; and the connectivity and geometry of the element neighbourhoods. Understanding the type of surface model assumed by the algorithm is helpful to abstract and generalise the techniques utilised to define direction, accumulation, and catchment so that they can be applied to other surfaces. Differences between the results of comparable algorithms determined on different surfaces may suggest improvements to both the algorithm and the surfaces.

4.3.1. Flow direction

Flow direction is a field dependent geomorphometric variable that indicates the direction that water will flow, provided momentum and pressure are insignificant (Gallant and Hutchinson, 2011). In In H&H models, flow direction algorithms are usually coupled with flow accumulation algorithms to form a flow routing algorithm. However, it is frequently possible to separate the direction and accumulation components of a flow routing algorithm. This subsection deals only with flow direction algorithms, flow accumulation follows in Subsection 4.3.2.

Flow direction is the path of steepest descent and is analogous to aspect. Flow direction can be calculated using formulae for determining aspect such as that described in Ritter (1987). However, flow direction and aspect are distinct due to the application of flow direction to flow routing. It is important to avoid loops in flow direction algorithms because loops would generate areas of infinite upstream area.

Point values for flow direction can be determined analytically if the terrain model is differentiable. However, in 2-D tessellated surface models flow direction is typically estimated for a finite but non zero area using a local neighbourhood operation. There are many flow direction algorithms for tessellated surfaces in academic literature; there are also some for networks, however, due to the dual nature of tessellations and networks, the difference is often a matter of perception and framing rather than one of substance.

Outside of tessellations, Mizukoshi and Aniya (2002) describe algorithms to generate flow lines and aspect from contours. Alternately, instead of routing cell to cell, flows can be routed directly from source to sink (Olivera et al., 2000). Sicilia and Judice (2010) describe techniques for animation of water flows in computer games using networks.

Directions can be defined in various ways and measured using various units. For instance, direction may be defined using geographic directions clockwise from north (or the y-axis); or mathematical directions anticlockwise from east (or the x-axis). The direction could be recorded in radians, degrees, or with a code derived from 8-bit binary encoding such as that in Figure 4.1. One benefit of 8-bit binary encoding is that all combinations of cells have a unique sum, which is useful for representing multiple flow directions with a single number.

Flow direction algorithms are usually applied to datasets that have been pretreated with a sink filling or breaching algorithm. Otherwise the flow direction results may need adjusting using soft hydrological conditioning. The edge cells of a flow direction raster must be treated as a special case. They may be calculated with a reduced neighbourhood, left undefined, assigned as sinks, or routed off the dataset.

The following discussion systematically classifies flow direction algorithms based on the algorithms' specification: what the algorithm purports to achieve, not the computational implementation. Tessellated flow direction algorithms can be categorised in a four dimensional matrix with each dimension representing a systematic classification of the algorithm. The four dimensions are listed below.

(1) Neighbourhood: Line adjacent, or line and point adjacent.

(2) Direction: restricted or unrestricted.

(3) Count: single or multiple; this is referring only to flow direction not flow accumulation, which can also be single or multiple.

(4) Geometry: Rectangular, hexagonal, triangular, irregular...

Further elements can then be added to meet certain objectives, such as stochastic modelling. Table 4.1 gives an example from academic literature (where such exists) of each class for rectangular raster DEMs.

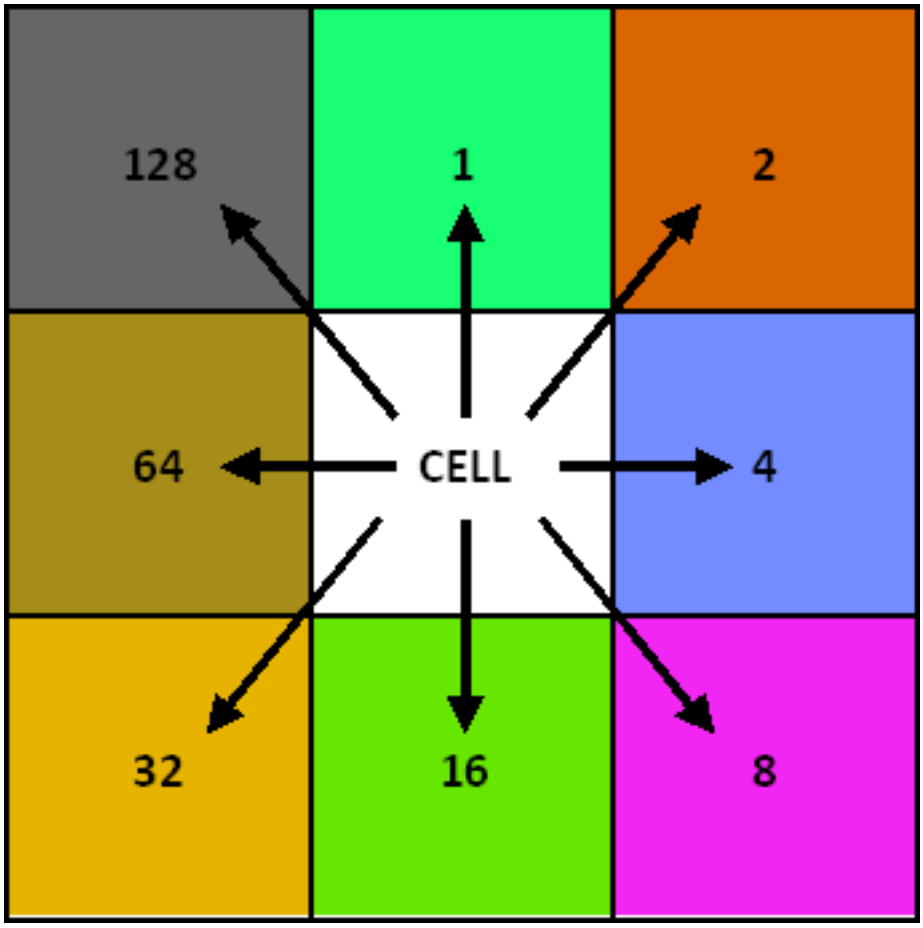


Figure 4.1.: *D*8 flow directions. The centre cell is assigned a number representing the direction of the steepest down slope neighbour. GIS software symbolizes the flow direction values using nominal colours to aid visual interpretation. The above colour scheme is used in Figure 4.3. Reproduced from Wright and Leonard (2012).

Within each class in Table 4.1 there are distinctive algorithms, which differ in their method for determining direction. Frequently, it is possible to apply these distinctive methods to classes other than the one they were described in, yielding a very large panel of possibilities from which to select the "best" flow direction. "Best" flow direction depends on desired use. For instance, dispersive algorithms such as $D\infty$ are better for determining flow accumulation whereas non dispersive such as *D*8-LTD are better for determining flow paths and hence catchment perimeter (Gallant and Hutchinson, 2011).

Single restricted

Single Restricted Flow Direction (SRFD) algorithms identify a single direction representing all flow passing through that cell. In addition, the directions are restricted only to those that correspond to the direction from the centroid of the cell in question to the

Table 4.1.: : Classification of flow direction algorithms with selected examples from academic literature.

	Line Adjacent	Line and Point Adjacent
Single Restricted	D4	$D8$
Single Unrestricted	-	$D\infty$
Multiple Restricted	M4	MD8
Multiple Unrestricted	MFM	MD∞

centroids of the cell's neighbours. The square tessellation example is shown in Figure 4.1. Restricted algorithms form connected networks and are, therefore, suitable for producing catchment boundaries. SRFD arrays are effectively Directed Tree Graphs (see Subsection 3.3.2. If multiple catchments are included they are forests.

The archetype SRFD algorithm for rectangular meshes is $D8$ (O'Callaghan and Mark, 1984). $D8$ considers all neighbours and direction is assigned to the steepest down slope neighbour. D4 is the equivalent single restricted flow direction algorithm where only neighbours adjacent across a linear boundary are considered. Some algorithms, motivated by computational speed, assign flow to the lowest elevation neighbour rather than the steepest down slope (Magalhàes et al., 2012).

In a cell centred mesh that has both point and edge adjacent neighbours the connectivity problem arises. In certain situations it is not clear whether an arrangement allows draining or is dammed. $D8$ implicitly assumes draining, D4 implicitly assumes damming but it is not obvious which is the correct approach without ancillary information.

$D8$ and D4 can be generalised to other geometries of tessellation: regular triangles, regular hexagons, and irregular tessellations. D6 is the hexagonal grid form of $D8$ (de Sousa et al., 2006; Wright et al., 2014). Triangular mesh equivalents of $D8$ and D4 would be D12 and D3 respectively. $D8$ analogues can be determined using irregular Voronoi diagrams. For instance, local difference run-off modelling with Voronoi cells is presented in Dakowicz and Gold (2007). The length of Voronoi edge in Voronoi $D8$ analogues can be used as flow width to do 2-D modelling (Tucker et al., 2001).

SRFD algorithms are highly approximated representations of flow fields, which has consequences when they are used to form flow accumulation fields or catchment boundaries. For instance, when SRFD algorithms are accumulated they cannot generally model dispersion.

There are distinct and predictable errors in $D8$ generated flow directions. The $D8$ algorithm has a direction bias because it can only select from eight possible directions, so any terrain features that naturally drain between these directions are modelled poorly (Tarboton, 1997). On some surfaces, these inaccuracies cancel out along the flow path,

preventing the flow from departing too far from the true location. However, the plane surface (Figure 4.2 A1) demonstrates how this is not always so. As flow descends along the plane's surface, the flow lines diverge farther and farther from the theoretical lines. $D8$ also tends to create parallel streams that do not converge. For example, the cone surface (Figure 4.2 A2), which contains areas of parallel lines and only localised convergence (Wright and Leonard, 2012).

There are several variations to the generic $D8$ algorithm intended to address directional bias. For instance, Orlandini et al. (2003) describe two methods that carry discrepancies of flow direction from the $D8$ method along the flow path known as $D8$-LAD (eight drainage directions, least angular deviation) and $D8$-LTD (eight drainage directions, least transversal deviation) that create non-dispersive connected networks. When the accumulated discrepancy reaches 45 degrees the direction is altered. The discrepancies carried forward by these methods are the difference between $D8$ and $D\infty$ directions (see Subsection 4.3.1). The difference between $D8$-LAD and $D8$-LTD is whether the discrepancy is angular difference or linear difference.; this difference is described in more detail in Subsection 8.5.2.

Fairfield and Leymarie (1991) proposed the Rho8 flow direction method that introduced a random factor to the steepest neighbour calculation so that the probability of selecting the steepest or second steepest downslope path has an expected value that is equivalent to aspect. Rho8 introduces randomness into the process by multiplying the elevation difference between the cell and its diagonal neighbours by a random factor, ρ_8 given as:

$$\rho_8 = \frac{1}{2 - \sigma} \tag{4.4}$$

where σ is a uniform random variable between zero and one. The expected value of $\rho_8 \approx 1/\sqrt{2}$. Compared to $D8$, if $\rho_8 < 1/\sqrt{2}$, the cardinal neighbour (non-diagonal) is favoured, if $\rho_8 > 1/\sqrt{2}$ the diagonal neighbour is favoured, if $\rho_8 = 1/\sqrt{2}$ the $D8$ solution is applied. Subsection 8.3.3 optimises Equation 4.4 so that ρ_8 is closer to $1/\sqrt{2}$.

Rho8 differs from adding random noise to the DEM because it results in no information loss and flows cannot travel uphill. However, flows converge randomly (Fairfield and Leymarie, 1991), and the results differ between individual runs. Figure 4.3, Row 5 shows examples of Rho8 flow direction arrays on planes, concentrative cones and helical surfaces. Unlike the $D8$ results (see Figure 4.3, Row 4), the Rho8 flow lines do not consistently flow in one direction. Some cells randomly select alternative directions, with this occurring at a frequency proportional to the gradient difference between the two options, thus creating a mottled appearance.

The random variable in Rho8's flow direction calculations produces non-deterministic behaviour. The randomness is non-repeatable, so results will differ between model runs. Different runs of Rho8 produce flow direction rasters with approximately the same relative proportions of selected flow directions, but due to the random factor, the calculated flow direction for an individual cell may change from run to run. Wright and Leonard (2012) evaluated *D*8 and Rho8 for determining catchment boundaries and found that if the results of multiple Rho8 runs are averaged the stochastic model can reveal hydrologically significant aspects of the topography that are not apparent from the application of a deterministic model, such as areas where bimodal behaviour is apparent.

A related issue with Rho8 is that flows on parallel paths merge and cannot separate, so downstream errors will accumulate, with more and more flow erroneously concentrated onto some cells (Costa-Cabral and Burges, 1994). Figure 4.2 confirms that the degree of flow convergence in Rho8 differs from the expected behaviour for a theoretical surface. In the plane example shown in Figure 4.2, Row 1, Column B, the original 18 flows converge to only five flows at the bottom of the plane. Convergence is a common occurrence in natural flows, but on plane and helical surfaces, convergence in Rho8 flow directions is caused by randomness, not underlying topography. Random convergence causes some downstream cells to receive inflated flow accumulation values (Wright and Leonard, 2012).

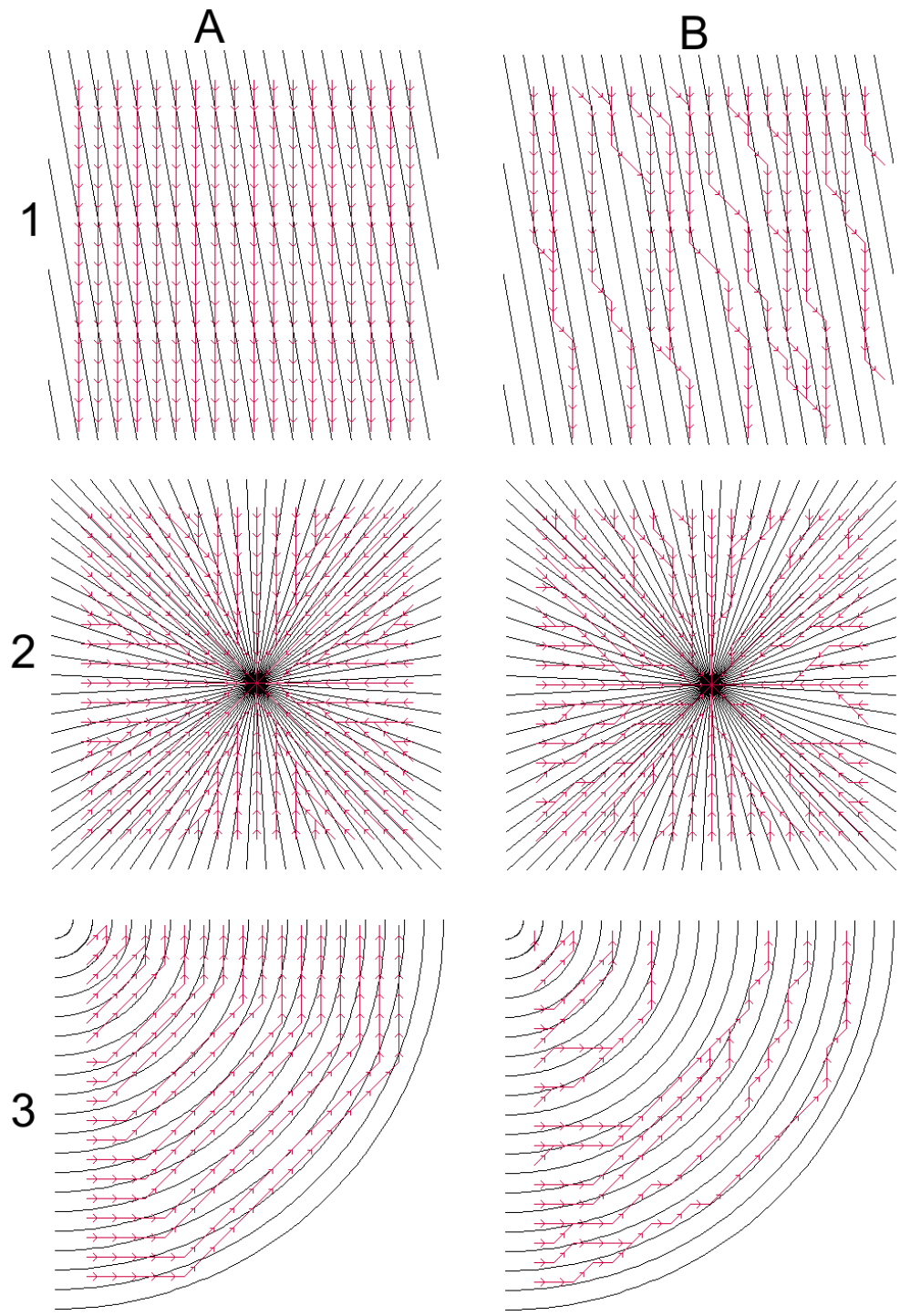


Figure 4.2.: Flow paths of $D8$ and Rho8 (shown in red). Theoretical results are shown in grey. Only the flow paths for the edge cells are shown. Column A represents $D8$ results. Column B represents Rho8 results. Row 1 shows results for a plane surface, Row 2: a cone surface, and Row 3: a helical surface. The surfaces are further described in Figure 4.3.

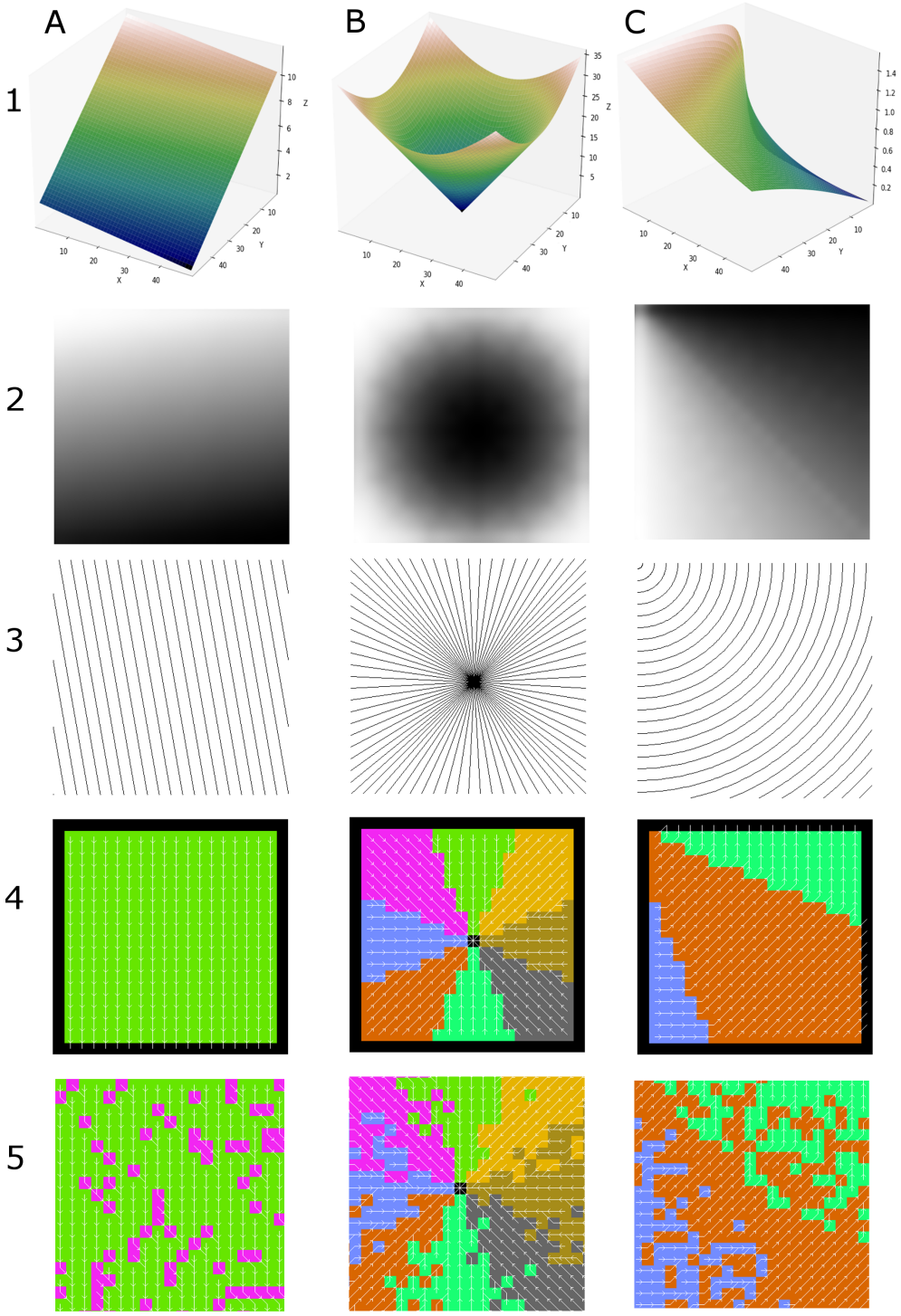


Figure 4.3.: Restricted flow directions on plane, cone surfaces, and helical surfaces. Each column presents results from a different surface. Column A is a plane that slopes downhill at a direction of 170º from the top of the diagram, Column B is a cone shaped pit, Column C is a helical curve descending anti-clockwise. Row 1 shows 3-dimensional representations of the surfaces. Row 2 shows raster DEMs of the surfaces where lower elevations are darker. Row 3 shows theoretical flow paths originating at evenly spaced points at the top of the surfaces. Row 4 shows *D*8 flow direct results. Row 5 shows Rho8 flow direction results. Rows 4 and 5 use the colour scheme in Figure 1 with white arrows overlaid to illustrate direction. Reproduced from Wright and Leonard (2012).

Single unrestricted

Single Unrestricted Flow Direction (SUFD) algorithms determine a single direction for the cell but it can be any direction. If $D8$ is the archetypal single restricted flow direction it is probably fair to say that $D\infty$ (Tarboton, 1997) is the archetypal single unrestricted flow direction. $D\infty$ defines flow direction as the aspect of the steepest down slope path on the set of triangles formed out of the cell and its eight neighbours. The slope is calculated for all 8 triangles and the steepest is assigned as the direction. $D\infty$ directions could also be resolved using only 4 neighbours; in fact, the Max Flux Method (MFM) proposed by Gruber and Peckham (2009) does something similar; however, MFM is a multiple flow direction. A generalised version of $D\infty$ for any geometry of tessellation is introduced in Subsection 8.3.2.

Unrestricted flow direction is generally determined by fitting one or several planes to the neighbourhood of the cell and determining the steepest direction on the plane(s). There are various plane fitting methods that could be applied, such as that used by DEMON (Costa-Cabral and Burges, 1994), box centred variations in Endreny and Wood (2001), plane fitting with least squares (Lee, 1991; Costa-Cabral and Burges, 1994); fitting quadratic trend surfaces, for instance using the Evans-Young formula (Shary et al., 2002); and Lea's Method (Lea, 1992). Methods that fit single planes to a neighbourhood of more than three points cannot, generally, fit exactly, so there are discontinuities at edges and potential for loops (Tarboton, 1997). $D\infty$ does not create loops because it fits surfaces to only three points, therefore, avoiding discontinuities.

Multiple restricted

Multiple restricted direction methods distribute flow only in the directions of neighbouring cells but may include more than one neighbour. Quinn et al. (1991) described such models on a regular rectangular mesh under the term MD8. MD8 may direct flows to one, several, or all downstream neighbours. Multiple direction algorithms resolve ambiguity in ridges and peaks (Peckham et al., 2009). Multiple restricted algorithms can readily be applied to other geometries.

Multiple unrestricted

Multiple unrestricted direction methods are represented by MD∞ (Seibert and McGlynn, 2007), which combines $D\infty$ with MD8; and the Mass-Flux Method (MFM) (Gruber and Peckham, 2009). MFM forms $D\infty$ like unambiguous facets with only the 4 edge adjacent neighbours to create 4 independent continuous directions. These can be summarised to

a whole pixel direction by finding the sum of the quarter-pixel flow vectors (Gruber and Peckham, 2009). MFM requires further conditioning to resolve crossing flow paths.

Flow direction on TINs can be taken as steepest downhill slope on the triangle surface (Liu and Snoeyink, 2005). Other examples of TIN flow direction include Palacios-Vélez and Cuevas-Renaud (1986), Jones et al. (1990) and Liu and Snoeyink (2005). TIN faces are effectively MD∞ generalised to irregular Voronoi tessellations because the TIN structures are generally the dual of a Voronoi tessellation. TIN models can distinguish between flows within and between triangles (Tucker et al., 2001). Triangle based flow routing flags as channels anywhere that two triangles slope together (Tucker et al., 2001).

4.3.2. Flow accumulation

Flow accumulation is the contributing area (see 4.1.3) of a cell in a tessellation. Flow accumulation is a kinematic simplification of physical hydrological process because it only considers elevation. Variables other than contributing area can also be accumulated such as slope and aspect (Heine et al., 2004). The relationship between the flow accumulation of a cell in a tessellation and Specific Contributing Area a as defined in Equation 4.1 is discussed further in Subsection 8.5.1.

Flow accumulation algorithms calculated on a grid raster typically require a flow direction raster as an input. Such algorithms usually determine the number of cells that contribute flow to a given cell, which can be presented as a count or as a normalized area by multiplying the count by the cell area. Flow accumulation rasters are often equalised for display purposes, as a small number of cells often end up having a much greater flow accumulation than the bulk of the cells (Wright et al., 2014).

Cell based flow accumulation algorithms perform two interrelated functions:

1. They define how flow is routed from one cell to another based on the flow direction.
2. They sum the flow reaching each cell in the flow accumulation raster.

Function 2. is commonly accomplished using a recursive flow climbing algorithm. In restricted flow direction algorithms, such algorithms can also assign pixels to catchments (Lin et al., 2008). Flow accumulation on large datasets can be difficult due to memory constraints because it is a regional algorithm, necessitating parallel solutions to break up the data. Gong and Xie (2009) describe data decomposition by catchment in multi-scale DEMs. The remainder of this subsection is concerned with function 1. i.e. how flow accumulation routes flow.

Flow routing on tessellated surfaces divides the accumulated flow of a cell between its neighbours in accordance to the flow direction of the cell. The sum of flow from one cell

to its neighbours cannot exceed the flow entering that cell (including the flow for that cell). Algorithms must specify whether the flow accumulation reported for a cell includes the cell's own area. A Single Flow Accumulation (SFA) assigns all flows to a single neighbour and will, therefore, create a one dimensional network. However, many grid and TIN-based models use a Multiple Flow Accumulation (MFA) that partitions flow from a single grid cell into one or more surrounding model elements creating a two-dimensional network (Kampf and Burges, 2007). MFAs can be limited to only two cells as in $D\infty$ or extended to all downhill cells (Freeman, 1991).

Routing of restricted flows

SRFD algorithms inherently assign flow entirely to one neighbouring cell, therefore, flow routing is trivial. The $D8$ and Rho8 flow direction algorithms are usually routed with SFA, where flow from each cell is routed to one or no other cells. SRFD algorithms do not model dispersion.

MRFD arrays are typically determined using a MFA, where all downstream recipient cells receive flows. MFA creates divergence, meaning flows spread as they descend. Divergence is an inherent characteristic on some surfaces. However, dispersion beyond the limit of geometric convexity is inconsistent with up slope area definition (Tarboton, 1997).

MFA can be distributed proportional to slope (Quinn et al., 1991) or proportional to slope to an exponent (Freeman, 1991; Quinn et al., 1995; Seibert and McGlynn, 2007). Adjusting the exponent fine tunes the behaviour of multiple MFA algorithms. At the extremes, exponent controlled MFA distributes flow evenly to all downstream neighbours or only to the most down stream, i.e. SFA $D8$. Endreny and Wood (2001) and Tarboton (1997) suggest that the MFA exponent should be left as an adjustable parameter. Quinn et al. (1995) vary MFA exponent based on distance from channel network.

Routing of unrestricted flows

In $D\infty$ flow accumulation is apportioned to the two neighbours closest to the direction of flow. The accumulation is apportioned proportional to the angular distance from the direction (see Subsection 8.3.1). $D\infty$ avoids excessive dispersion but is anisotropic because there is more dispersion away from 45°. An alternative to angular offset used in $D\infty$ is linear offset (Orlandini et al., 2003). The difference between angular and linear offset influences the extent to which the steeper cells are favoured. Linear offset places relatively more flow toward the steeper neighbours.

MFM essentially applies a MFA to multiple $D\infty$ flow directions determined on linear neighbours. In MFM, flow accumulation is apportioned from the quarter pixels defined by two adjacent linear neighbours to the recipient whole pixel proportional to the decomposition of the flow vector into its x and y components (Gruber and Peckham, 2009).

Other flow accumulation methods

The multiple and single flow accumulation algorithms discussed above model flow as a line from one cell into another. DEMON (Costa-Cabral and Burges, 1994)) is an alternative to the linear assumption. DEMON models flow as a two dimensional flow tube. DEMON areal flow routing has separate algorithms for up and downstream flows. In DEMON downstream flow, every cell is modelled individually to its outlet or sink. Costa-Cabral and Burges (1994) proposed DEMON with surface fitting but it could easily be applied to other flow direction algorithms. Another example of flow routing that does not fit the MFA/SFA model is Lee's method, which models flow as a rolling ball, recording where flows enter and exit each cell, starting from the centre of the cell they originated in Lee (1991). Lee (1991) uses plane fitting for flow direction. However, a variation of Lee's method is used in the commercial software CatchmentSIM Catchment Simulation Solutions (2008) that does not use plane fitting to calculate directions, illustrating how flow direction and accumulation are distinct steps that can be abstracted from each other.

A two dimensional method that captures the effects of buildings at a scale finer than the modelling scale is presented in Chen et al. (2012). Their method captures flow paths on multiple vertical levels where the effect of buildings is incorporated in other levels, effectively representing finer scale effects where required. Their method uses a cellular automata method to identify areas were buildings create flow interactions that need modelling across multiple levels and also allows the possibility to model factors such as underpasses. For flow algorithms on TINs, refer to Vivoni et al. (2004) and Liu and Snoeyink (2005).

Analytical flow accumulation for assessment

Specific catchment area (a), can be determined from first principles analytically on convergent and divergent cones, and inclined planes. Formulas for accomplishing this are given in Gallant and Hutchinson (2011). Outside these mathematical shapes, methods of assessment generally consist of comparison between methods and with intuitive expectation of results. More computationally intensive methods for determining flow accumulation that are useful for accuracy assessment include stream tubes (Orlandini et al., 2003) and a differential method (Gallant and Hutchinson, 2011).

Gallant and Hutchinson (2011) develop the Orlandini et al. (2003) and Orlandini and Moretti (2009) contour and flow line network approach into a differential equation method that can analytically determine a anywhere an equation can be defined from the DEM that forms a surface that has continuous first derivatives and piecewise continuous second derivatives. Surfaces built with quadratic splines have piecewise continuous 2nd derivatives, which allow the numerical solution of a (Gallant and Hutchinson, 2011). Gallant and Hutchinson (2011) define rate of change of a along a slope line (l) as a function of a and plan curvature (k_c):.

$$\frac{\partial a}{\partial l} = 1 - ak_c \tag{4.5}$$

Integrating along the slope line yields a at every point.

Gallant and Hutchinson (2011)'s method is accurate in the sense that it is accurately representing the mathematical surface, not that it accurately represents the flow lines of the real surface. The result is removed from the real world both by the limitations of the accuracy of the DEM and the accuracy of the mathematical surface that has been fitted to it (Gallant and Hutchinson, 2011). In addition, further approximation comes from the numerical solution of differential equations. However these can be managed to negligible levels with care (Gallant and Hutchinson, 2011).

A selection of flow routing methods was described and evaluated on mathematical surfaces in Pan et al. (2004): Single Flow Direction (SFD) (O'Callaghan and Mark, 1984), Bi-Flow Direction (BFD) (Tarboton, 1997), Multiple Flow Direction (MFD) (Quinn et al., 1991), and MFD*, which includes a minor alteration of the weighting of diagonals of MFD (Wolock and Mccabe, 1995). Pan et al. (2004) found MFD to be the most accurate. Erskine et al. (2006) compare five flow direction/accumulation methods ($D8$, M8, MFD, DEMON, and $D\infty$) and explore differences between them, eventually determining trends between multiple and single flow algorithms. Erskine et al. (2006) found that relative differences between accumulation values estimated using single- and multiple-direction algorithms increased with decreasing grid cell size. Relative differences were greatest along ridges and side slopes, and differences decreased where the terrain became more convergent. Erskine et al. (2006) recommend Multiple-direction algorithms on undulating terrains because they allow for flow divergence. The sensitivity of flow routing methods to elevation errors is studied in Endreny and Wood (2001).

4.4. Hydrological conditioning

In order to create DEMs with realistic drainage properties, modellers sometimes apply hydrological enforcement techniques during data interpolation, or condition the surface after interpolation to improve hydrological behaviour. Callow et al. (2007) make a distinction between hard hydrological conditioning, which modifies the DEM and soft hydrological conditioning that relies on algorithms to adjust the outputs of hydrological analysis without affecting the DEM. Hard hydrological conditioning can compromise later analysis (Callow et al., 2007). However, so can soft hydrological conditioning if later analysis involves both altered and unaltered inputs, for instance adverse flows could be generated when examining channel gradients. This section divides hydrological conditioning into two broad categories, pit removal (Subsection 4.4.1) and ancillary data (Subsection 4.4.2). Hydrological Conditioning on TIN surfaces (Subsection 4.4.3) and in urban environments (Subsection 4.4.4) are discussed separately.

4.4.1. Pit removal

Pit removal involves removing pits (see Subsection 4.2.1) and flat areas from the dataset. Pit removal in the field of image segmentation is described as prevention of over segmentation (Moga and Gabbouj, 1998). For either steepest path or immersion catchment techniques to work, datasets need to be lower complete, that is, each cell must have a lower neighbour unless it is an edge cell or a genuine local minima (Roerdink and Meijster, 2000). Sinks are removed from DEMs on the assumption that they are rare in nature (Hutchinson, 1989; Goodchild & Mark, 1987) and therefore an artefact of data collection or interpolation. If pits are not real features of the landscape, they should be removed from the model or accounted for during the flow direction, flow accumulation process.

This section describes techniques for removing pits from regular arrays, although many of the techniques described could also be applied to irregular arrays. For further information, see Kenny et al. (2008) for an overview of pit removal methods and Lindsay and Creed (2005b) for an assessment of the effect that filling methodology has on hydrological outputs. It is important to recognise that adding higher resolution to tessellated surfaces does not necessarily resolve sinks even if the shape of the surface is known to an extent that the sink is not present. This is because sinks can be created by the interplay of the tessellation geometry and the surface topography. To resolve such sinks, there needs to be some process of breaching or filling.

Pits are generally assumed to be spurious. However, genuine sinks play an important

role in landscape evolution so should not be removed without considering whether they are real (Temme et al., 2006). Genuine pits are fairly common in the built environment; the development of land creates sinks, which are then typically drained into underground pipe networks. For obvious reasons, these sinks should not be removed for the purposes of modelling urban hydrology (Wright and Leonard, 2012). It is, therefore, necessary to distinguish actual and spurious depressions before removal. Lindsay and Creed (2006) examine some methods to do this. The method of Hutchinson (1989) effectively distinguishes between real and spurious pits.

Arnold (2010) shifts the assumption away from pits always being false and resolves them on the assumption they are real and need to flood and outlet. Arnold (2010) also presents arguments for determining the likelihood of erroneous pits in a DEM by statistical analysis. Monte-Carlo modelling has also been used to determine if sinks could be the result of data inaccuracy (Zandbergen, 2010; Lindsay and Creed, 2006).

Several techniques for removing pits are presented in the literature. The most prominent techniques are filling and breaching. Filling raises the elevation of cells within the basin formed by the pit creating a flat area at the elevation of the basin pour point (O'Callaghan and Mark, 1984; Meisels et al., 1995). Breaching lowers the elevation of cells on the perimeter of the basin allowing flow to escape.

Planchon and Darboux (2002) present an efficient filling algorithm based on flooding and removing excess water. Jenson and Domingue (1988)'s $D8$-topaz software utilizes a filling technique. Wang and Liu (2006) demonstrate an $O(NlogN)$ efficient filling technique. Magalhàes et al. (2012) describe an efficient algorithm that combines sink filling with flow routing. An alternative filling method uses ancillary data in the form of known stream channels as pour points if such pass through the pit's basin (Mark, 1988).

Martz and Garbrecht (1999) applied breaching to grid DEM. Soille et al. (2003) describe a breaching technique, called carving which creates sloped paths in the DEM to allow pits to outlet and discusses resolving flat areas. Breaching can also be effected during flow accumulation as a soft hydrological conditioning method leaving the DEM unaltered. Fairfield and Leymarie (1991), Chorowicz et al. (1992), and Chou et al. (2004) all take the soft breaching approach. Soft breaching can be as simple as changing the direction of flow from the pit to the lowest elevation cell on the basin's catchment. Flow reversal techniques do not usually adjust the DEM to match the reversal so will create adverse flows. Lin et al. (2008) apply a fuzzy logic rule to flow direction pit removal taking into account distance to the outlet and depth change between a cell inside and outside the basin. Kenny et al. (2008) describe Iterative Enhanced Flow Direction Grid (EFDG) pit removal using flow direction altering.

Filling assumes pits are caused by local underestimation, breaching assumes local overestimation. Filling some pits and breaching others based on the source of the error would be ideal. However, this approach is impractical as the source of the error is generally unknown. Pit removal techniques can be optimized for impact reduction (Lindsay and Creed, 2005a). Soille (2004) describes a mixed filling and carving approach that minimizes the sum of differences between processed and unprocessed cells. Generally breaching has less impact than pit filling. Applying the least impact approach to each individual depression is recursive and hard to implement (Lindsay and Creed, 2005a). Martz and Garbrecht (1999) propose a method by which breaches are applied up to a maximum horizontal distance and the remainder is filled. The intention of Martz and Garbrecht (1999)s technique is to breach only in areas that are blockages in a defined watercourse.

Flat areas within a DEM, whether from filling, extrapolation, or source data need to be assigned flow directions either by imposing relief or by algorithmically resolving flow direction. Flat areas can be resolved using the nearest resolved cell method (Tarboton, 1997) but this often creates unrealistic parallel flows. Garbrecht and Martz (1997) create a converging technique based on a gradient from upslope and gradient to downslope. Kenny et al. (2008) resolve flats in EFDG by extending surrounding flow direction into flat areas without altering the DEM. Filling can also leave behind slopes if required for drainage. Pem4pit filling (Grimaldi et al., 2007) considers geomorphological processes of erosion and uplift. Pem4pit uses a continuity of mass equation within the filling process to create more realistic surfaces compared to flat planes from traditional filling techniques.

4.4.2. Ancillary data

Hydrological conditioning is often accomplished by incorporating ancillary data into the surface model. Similar to pit removal, ancillary data can either be incorporated through the interpolation process, as hard conditioning of the DEM after interpolation, or using soft algorithms.

A common ancillary data technique is stream burning. Stream burning lowers the cells of a DEM in the vicinity of a known stream (Maidment, 2002). For a summary of stream burning methods, see Saunders (2000) and Jones (2002). Enhanced stream burning procedures include AGREE surface reconditioning of riparian zone (Hellweger and Maidment, 1997), floodplains (Getirana et al., 2009), exponential stream bed (Saunders, 2000), power law (Getirana et al., 2009), priority-first-search weighted-graph algorithm (Jones, 2002), and adaptive stream burning that only enforces drainage properties where there is significant disagreement between known and derived networks (Soille et al., 2003). Renssen and Knoop (2000) apply stream burning to lakes, distinguishing between lakes that are

part of drainage to the sea and those that are not. Turcotte et al. (2001) added a Digital River and Lake Network (DRLN) to flow directions then altered the DEM as a function of distance to DRLN.

Stream burning can be accomplished as a soft algorithm by altering the flow direction, see for example the method of Kenny and Matthews (2005), which was extended to resolving flats in Kenny et al. (2008). In a different approach, the commercial software described in Catchment Simulation Solutions (2008) uses vector overlays to model channel positions, where array cells behave differently if they are overlapped by the vector features. Ancillary data can also be introduced during interpolation, such as streamlines and break lines in ANUDEM (Hutchinson, 1989). Hengl et al. (2008) use geostatistical regression to incorporate ancillary data into the interpolation process.

Other elements of the built environment with known position and hydrological characteristics can be burnt into the raster DEM in processes similar to stream burning. Surface features such as roads, drainage ditches and kerb and channel that are included this way can capture and divert large volumes of surface water, despite often being smaller than the resolution of the raster DEM. In such cases, the features being modelled need to be exaggerated sufficiently to be represented in the output before "burning" into the elevation data (Wright and Leonard, 2012).

Despite generally being of fine spatial resolution, DEMs produced from LiDAR data still require hydrological conditioning. LiDAR data are often captured at a spatial resolution that makes it possible to resolve road drainage features, but this is not always the case. This can have an adverse effect on road burning processes as inaccurately positioned ancillary data such as road centrelines could potentially create undesirable artefacts when combined with LiDAR elevation data containing information on drainage features (Goepfert and Rottensteiner, 2009).

Duke et al. (2006) adapted stream burning techniques to model roads (Road Enforcement Algorithm (REA)) and canals (Canal Enforcement Algorithm (CEA)), then combined REA and CEA to make RIDEM which alters flow direction to capture anthropological, scale hydrological controls without affecting the DEM. Roads can be incorporated into 2.5D GIS DEMs by combining vector data and LiDAR data using an adaptive snake (Goepfert and Rottensteiner, 2009). Using dynamic contours, vector object data can be matched to the implicit object information in the LiDAR surface. Roads can be recognised based on their low intensity texture and gradient changes.

Other examples of incorporating ancillary data are: Schäppi et al. (2010), who introduce measurements of river cross sectional data into the DEM, and Lindsay et al. (2008), who position gauge data to match derived stream networks using topology and names. This

technique could be applied to fit stormwater drains to channels.

Callow et al. (2007) study the effect of stream burning on hydrological outputs. Getirana et al. (2009) focused on stream burning and parameters for floodplain burning, attempting to achieve acceptable networks with minimum alteration of a DEM. ANUDEM was the best performing algorithm. Grimaldi et al. (2004) show that ANUDEM mimics known river shape with the least disruption to the DEM compared to stream burning, AGREE, and Grimaldi et al. (2004)'s physically based model.

4.4.3. TIN hydrological conditioning

TIN data sets are also affected by spurious sinks. Liu and Snoeyink (2005) discuss filling sinks in TINs. Sinks are filled sequentially based on one of several proposed criteria, area of sink, volume of sink, and spill time, which is volume divided by contributing area. The order in which pits are filled and catchments dissolved is recorded allowing different limits to be applied to the time function thus generating greater or smaller levels of pit resolution from the same run of the algorithm (Liu and Snoeyink, 2005). Their experiments showed catchment sequence defined by area produced results closest to manually defined catchments. These sequential rules are comparable to height to be filled limitations used in some grid algorithms, for example, Lin et al. (2008) and could be applied to other data structures. Sinks in polyhedral terrains are addressed using elevation alteration of vertices in Silveira and van Oostrum (2007). Koch and Heipke (2006) discuss introducing ancillary 2D GIS data into a 2.5D TIN in a semantically correct way.

Vivoni et al. (2004) develop a hydrological similarity interpolation, which use TWI (topographical wetness index) instead of slope for preservation. This method is a form of hydrological enforcement in the interpolation of TINs from DEMs. Vivoni et al. (2004) identify three ways of making TINs, traditional, hydrographic (includes basins, rivers, retains slopes), and hydrological similarity. Hydrological similarity controls the resolution of the DEM to represent areas of high topographic wetness index with higher resolution. Therefore, using hydrological similarity one can preserve hydrologically significant features with fewer points. Hydrological similarity is further developed and tested in Vivoni et al. (2005b). Hydrographically constrained triangulation includes river networks as hard break lines and catchment boundaries as soft break lines. The river and catchment information can be calculated from a DEM, generalized, and then included in a TIN.

4.4.4. Urban hydrological conditioning

Urban surfaces have a number of characteristics that need to be considered in order to accurately model the direction of flow in urban areas. Urban surfaces are discussed in detail from a hydrological standpoint in Appendix B.3. Assumptions regarding surface flows that are generally sound in undeveloped areas do not always apply to urban areas. Therefore, hydrological conditioning must be adapted for urban environments.

Urban surface features such as engineered drainage channels and stormwater reservoirs can have large, but highly localised, effects on a catchment's surface water hydrology by redirecting water from natural pathways despite only being metres or centimetres in width. Due to artificial channels, flows may appear to run along contour lines, and surface flows may arrive from and disperse to multiple different directions. An urban surface model must be capable of capturing important fine scale land surface features such as roofs, roads, ponds, pumps, drainage ditches, impervious surfaces, and kerb and channels (Panday and Huyakorn, 2004; Yeh et al., 1998). Spatial models can model fine scale features by using very fine grids, using unstructured grids with variable density, coupling grids of different density, by considering sub cell size topography, by multi-scale modelling, or by utilising special hydrological conditioning methods.

To be effective in urban areas hydrological models also need to consider interactions with pipe networks. Pipe networks increase complexity for hydrological models in urban areas because they introduce small subcatchments, genuine pits at the interfaces with pipe network inlets, and intermittent overland flow paths (Smith et al., 2006). In addition, the pipe network may become overloaded during heavy rain and stop receiving flow and become a flooded pit. Meierdiercks et al. (2010) found using computer modelling that drainage infrastructure is actually more important than impervious surfaces in predicting rainfall response in urban catchments.

For effective urban hydrological conditioning, pipe networks must be considered in order to determine if pits are genuine. However, false sinks from, for instance, bridges being treated as ridges still need to be removed. Software that combines 2-D overland flow with 1-D pipe modelling are frequently termed 1-D/2-D models (Bandaragoda, 2008). 1-D/2-D models are a form of hydrological conditioning of the 2-D model. In a 1-D/2-D model some pits are the interfaces between the 2D surface and the 1D pipe network.

The implementation, operation, and results of algorithms that determine variables and objects from surface models, such as the geomorphometric algorithms and hydrological analysis outlined in this chapter are influenced by the characteristics of surface models

that were described in the previous chapter. In the interests of good modelling practise, as introduced in Section §2.1, it is necessary to be as clear as possible about how these interactions between surface and algorithm affect model results. However, the interaction can also go the other way, just as the surface model will affect the analysis algorithm, the algorithm and the modeller's intentions may affect the surface model. Part II will describe a surface model that has characteristics that allow modellers to adapt the scale and geometry of a tessellated surface model to minimise error.

Part II.

The Regular Hierarchical Surface Model

In this Part, questions raised in Chapter 2 regarding modelling scale in surface models are answered by assembling selected components of the surface modelling framework constructed in Chapter 3 into a Regular Hierarchical Surface Model (RHSM), that in Part III will be applied to the geomorphometric hydrological analysis described in Chapter 4. Chapter 5 discusses the conceptual model of the proposed RHSM, Chapter 6 details the mathematics underpinning the surface, and Chapter 7 covers aspects of the computational model.

5. Conceptual Regular Hierarchical Surface Model

This chapter documents the conceptual model of the Regular Hierarchical Surface Model (RHSM) and explains its rationale. As such, it forms a conceptual bridge between the background information in Part I and the mathematical (Chapter 6) and computational (Chapter 7) details of the new surface model.

Section §5.1 identifies and expands on the questions of scale that were encountered in the literature review and proposes that the questions of scale should be addressed using multi-scale surface modelling. Section §5.2 discusses the advantages and disadvantages of modelling spatially distributed phenomena by dividing space into a regular discrete mesh and proposes that forming multi-scale models by subdividing scale using a regular discrete hierarchy has similar advantages.

Section §5.3 illustrates the benefit of hierarchical referencing of multi-scale spatial data in implicit pyramids. Section §5.4 proposes adapting the Level Of Detail (LOD) models used in computer graphics to capture scale dependence in distributed models of geographic data. Section §5.5 outlines the RHSM.

5.1. Questions of scale

This section identifies the issues of scale dependence that were discussed in a variety of contexts in Section §2.3 and elsewhere in Part I. This section proposes adding a branch to the modelling decision tree presented in Section §2.1. The new branch represents the option to model the variation of parameters with scale.

Philosophers from the time of Aristotle have recognised and described the fallacy of composition (Aristotle, 4th Century BCE). It is a fallacy of composition to claim that what is true of the parts is necessarily true of the whole. The complement of the fallacy of composition is the ecological fallacy: the process of inferring the characteristics of individuals from data about aggregates (Robinson, 1950).

A trivial example of the fallacy of composition is the claim that every person in New Zealand weighs less than an elephant, therefore, all people in New Zealand collectively weigh less than an elephant. This is clearly a fallacy. However, not all such claims are false. For instance, the claim that all people in New Zealand collectively weigh more than a mouse follows logically from the observation that all people in New Zealand individually weigh more than a mouse. The extent to which the properties of the parts can be applied to properties of the whole varies depending on the nature of the properties, and the objects or fields to which they apply.

In Section §2.3, the fallacy of composition was encountered in reconciling measuring and sampling scale with computational scale within hydrological models. In addition, it was recognised that many geomorphometric parameters, such as slope, change with spatial resolution. Therefore, it is necessary to describe surfaces with geomorphometric parameters that approach a limit when varied by changing spatial resolution.

Measurement of objects with fractal properties also reveals variation of parameters with scale (see Section §2.3 and Section §4.1). For instance the length of the perimeter of an object with fractal geometry, such as a coastline, varies depending on the resolution of measurement (Mandelbrot, 1967). Goodchild (2001) observed that a variable is scale dependent if a length is inherent in the definition of the variable. Therefore, many measured values are only relevant for a given scale.

When modelling a process in a GIS there are three interrelated questions of scale that arise: Do variables change with scale? And if so, how do you integrate variables (or processes) to form a generalised value (or process)? From these questions arises the third; if model outputs are affected by scale of measurement, what scale should we use?

Multi-scale data structures provide a mechanism to deal with issues of scale and agglomeration by explicitly modelling variables that vary with scale. Accommodating scale dependent variables in geographical models effectively adds a new layer to the modelling decision tree tree illustrated in Chow et al. (1988). The new layer asks whether the model should account for variation with scale (see Figure 5.1).[1]

The extra branch of the decision tree could be applied to any of the model types depicted as leaves in Figure 2.1. However, Part III considers scale variation in deterministic, distributed, steady flow hydrological models. Distributed modelling is the form of model which is most likely to vary with spatial scale. It is difficult to conceive of multi-scale lumped models that do not imply that the finer scale models are in fact distributed. Stochastic and unsteady models are avoided initially to simplify the modelling process

[1] Alternatively, due to the fact that multi-scale models are effectively both lumped and distributed, the decision to include scale variance in a geographic model could be conceptualised as an additional option on the spatial variation branch: lumped, distributed or multi-scale.

because, as Chow et al. (1988) indicated, it is difficult to capture all types of variation in a single model. By focusing on models that vary only spatially and with scale, the implications of scale dependent modelling can be isolated and investigated.

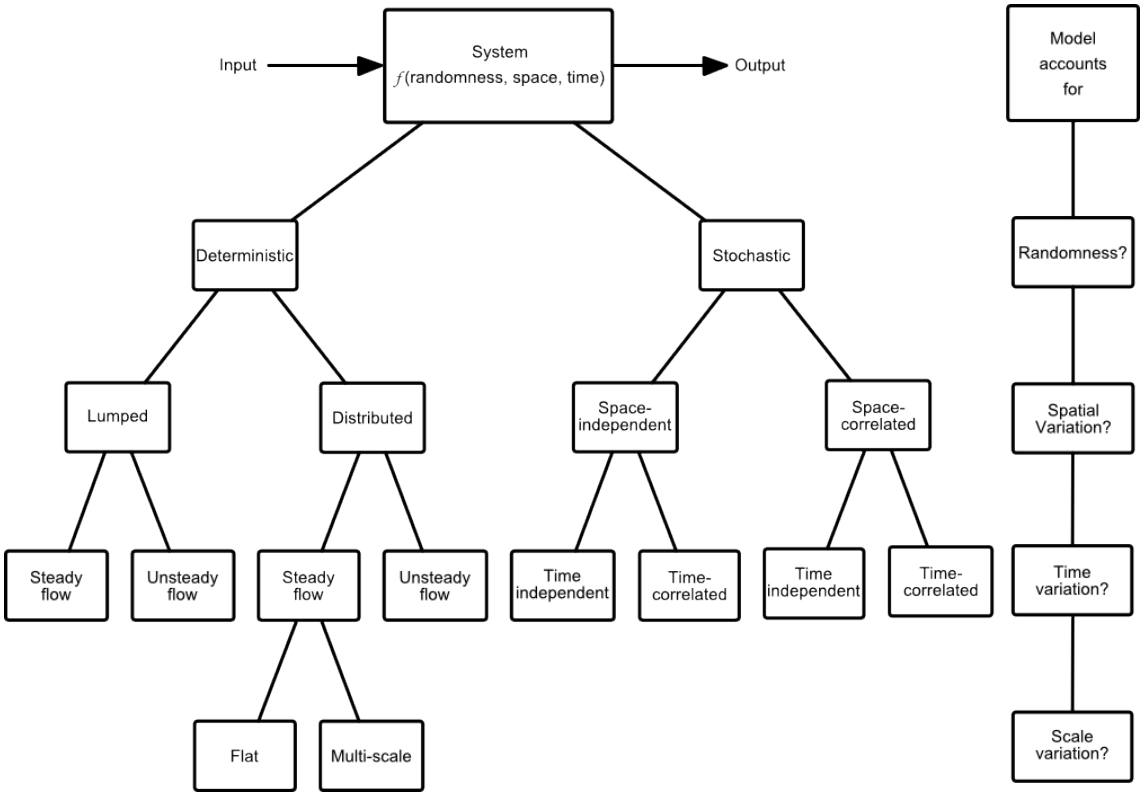


Figure 5.1.: Modelling decision tree with an additional option to account for scale variation.

5.2. Regular discrete subdivision of space and scale

This section reiterates the advantages of subdividing space in distributed models using regular geometries (Section §3.5) and implicit data models (Section §3.7), and extends the concept to the discretisation of scale. It identifies a regular hierarchy of regular tessellations as an efficient way of subdividing space and scale because such hierarchies have similar shapes, fixed aperture, congruent centres, and aligned boundaries (Section §3.8).

5.2.1. Regular discrete space

Not withstanding the Planck constant[2], the real world as experienced by humans is continuous. However, in computer models of fields, it is convenient to subdivide space into discrete entities within a tessellation or network graph. Fields can be modelled continuously using equations that interpolate values between discrete data points. However, surface modelling by discretisation of space is often more achievable, which is why numerical techniques, for instance, often solve continuous equations using discrete approximations. Even vector representations of objects in space do not typically represent location with analogue precision due to the underlying grid implicit in the resolution used to store the coordinates of vertices (Subsection 3.7.3).

A 2-D square raster is a typical example of a regular subdivision of space. A regular point set is created by sampling a surface at set intervals in linearly independent dimensions(x, y).If these sample points are treated as generators for a degenerate Voronoi tessellation, a regular rectangular tessellation is formed. The tessellation is square if the sample distance in x equals the sample distance in y. As discussed in Section §3.5, other regular point sets can be generated by tilting the axes, and changing the sampling distances.

Subdividing space with a regular geometry is advantageous for geographical modelling because it reduces computational overhead for storage and analysis due to the implicit position and shape of regular meshes. Algorithms designed to perform analysis on regular meshes can be simpler than those designed for irregular meshes due to the constant connectivity and adjacency of cells within a regular mesh. In this regard, one particularly useful geometry is the isotropic hexagonal Voronoi diagram. Hexagonal tessellations have no point adjacent neighbours, which resolves the connectivity problem, and equal distances to all neighbours (see Subsection 3.5.3).

A counter argument in favour of irregular geometries is that irregular geometries can alter their resolution spatially to generate variable density sampling. However, irregular geometries require explicit position and shape to be stored. The storage and computational overhead for modelling with irregular meshes is greater than for regular meshes because each cell has unique values for its area, number of neighbours, length of edge shared with each neighbour, and distance to each neighbour. Subsection 5.2.2 describes how regular hierarchical division of scale allows variable density meshes to be formed from regular geometries.

[2]The Plank constant, 6.626×10^{-34} joules, is the energy of a quantum of electromagnetic radiation divided by its frequency. Quantization of energy implies that only certain energy levels are allowed, and values in between are forbidden; therefore, at extremely fine scales, energy is not continuous.

Another argument in favour of irregular geometries is that they avoid the accumulation of errors, known as geometric bias that affects regular structures (Holland et al., 2007). If regular geometries are used for modelling, it is judicious to use geometries that minimise geometric bias and have geometric characteristics that are suitable for the processes that will be represented on them. One method to mitigate geometric bias is to repeat the analysis with rotated arrays, and/or adjusted origin. Chapter 9 investigates hexagonal, rectangular, and triangular sampling for forming regular surface models. Each of these tessellations may have advantages within specific problem domains.

5.2.2. Regular discrete scale

The capacity for subdivision with a regular structure is not unique to the dimensions of Euclidean space. For instance, time and numerical attribute data can be treated as a dimension and modelled with a regular structure. A model that allows parameters to vary with scale effectively treats scale as dimension, which, like space, can be modelled continuously or discretely. If modelled discretely, scale too can be subdivided with regular or irregular structures.

A model with continuous scale variation provides parameter values for any resolution; a defined and solvable scaling function could provide this. Alternately, discrete scale models provide only a subset of possible scales. The range of scales in a discrete system can be described by a sequence of values representing the area of an individual tile in each successively coarser scale relative to the area of tiles in the finest or "base" scale. These scale ranges effectively describe pyramid layers or hierarchies. This suggests a number of possible classes of discrete scale ranges. Discrete scale ranges could have no pattern and thus be irregular or the scale range could conform to a regular pattern.

For instance, a linear discrete scale range would have cell sizes equivalent to:

$$1 + ni \, \{i \in \mathbb{Z} \geq 0\} \tag{5.1}$$

where $n \in \mathbb{R} > 0$.

An exponential discrete scale range could conform to:

$$i^n \, \{i \in \mathbb{Z} \geq 1\} \tag{5.2}$$

or:

$$n^i \{i \in \mathbb{Z} \geq 0\} \tag{5.3}$$

The hierarchies formed by linear scale ranges such as Equation 5.1 would not generally be aligned or congruent. The hierarchies formed by Equation 5.2 could be made to be congruent but not generally aligned. The hierarchies that are formed by Equation 5.3 can be made to be aligned and congruent by choosing values of n equivalent to the aperture of self-similar hierarchies of the regular tessellations. The lowest value of n that can generate aligned congruent self-similar hierarchies depends on the geometry of the tiles; this value is 9 for rectangular tessellations, 7 for hexagonal, and 4 for triangular.

The self-similar, aligned, congruent hierarchies have characteristics that are distinctive to their geometries. For instance, the correspondence between neighbourhood and hierarchy can only be achieved for all three regular geometries by being flexible with the definition of neighbourhood. The rectangular geometry needs to include the point adjacent neighbours, whereas, the triangular geometry only includes the line adjacent neighbours. In addition, the aperture-7 hierarchy for hexagons is only somewhat self-similar and approximately congruent.

Congruent hierarchies make aggregating values easier and also make it possible to form variable density "realisations" (see Subsection 5.4.2) because cell boundaries are always congruent with finer resolution neighbours. Aligned hierarchies, with a single central child cell that is co-located with the parent, ensure that there is a clearly defined fine-resolution equivalent of the parent cell and create a correspondence between the geometry of a cell's neighbourhood and the geometry of the self similar unit, which facilitates hierarchical indexing and identifying neighbours.

The lower the value of n the denser the range of resolutions achieved, creating what could be described as a fine resolution of resolutions. An argument against schemes based on self-similar hierarchies is that the scale resolution is comparatively coarse. This objection is evaluated in Chapter 9.

A Regular Hierarchical Surface Model is a regular tessellation (and its dual network) of regular polygons, which employs an exponential sequence of scales (Equation 5.3). It may seem naive to reduce the complexity of the multi-scale world to such a simple scheme. However, the simplicity of the raster data model has not prevented its widespread application. Indeed, the simplicity of the raster data model has encouraged many complex operations to be built on its elementary scaffold.

5.3. Hierarchical referencing and pyramids

This section discusses hierarchical referencing and forming pyramids. Hierarchical referencing and pyramids are related because cells can be referenced and stored using the same groupings as are aggregated into pyramids. Hierarchical referencing efficiently organises the values required to form the agglomeration and implicitly associates values between different resolutions without storing links.

5.3.1. Hierarchical referencing of spatial data

As discussed in Section §5.2, the regular division of scale facilitates implicit hierarchy in the same way that the regular division of space facilitates implicit position and shape. Referencing systems can exploit this hierarchy to organise spatial data in a way that facilitates tiling and memory management.

Regular subdivision of space can be represented in linear arrays using space filling curves such as the HIP referencing system (Middleton and Sivaswamy, 2001). Linear storage using the HIP index is implicitly hierarchical because it does not require pointers. Another way of conveniently storing a nested hierarchy is in a tree data structure where the branches represent the decomposition of space into finer parts via a regular subdivision (Section §3.7). An advantage of the tree structure is that not all branches need be present, therefore, there is potential for sparse or variable density surfaces. However, explicit tree structures also require some overhead due to the need to direct branches to appropriate memory locations.

Storing values in tree structures in secondary memory demands a significant number of disk reading operations. The number of operations can be reduced by tiling, where rather than placing individual values in memory, arrays of values that are spatially adjacent can be stored in a tree structure (Platings and Day, 2004). This need not sacrifice the integrity of the hierarchical structure because the array can employ an equivalent hierarchical referencing system to the tree (see Section §6.2). Section §6.1 details the hierarchical referencing system used by the RHSM, which is an adaptation of the HIP system applied to both tree and array structures.

5.3.2. Pyramids and scaling functions

A pyramid data structure consists of a set of spatial representations that have the same extent but successively coarser resolutions. Therefore each subsequent layer has fewer

cells. Hierarchical referencing systems help create pyramid data structures because the values that need to be aggregated to form the coarser resolution share a common reference and can be made adjacent in system memory. In addition, a hierarchical referencing system easily relates values between levels of resolution because spatially coincident cells in different layers of the pyramid will share the same index except that the coarser resolutions will have shorter references (see Section §6.1).

If the value of a variable is represented in a model at more than one spatial scale (multiscale), a scaling function presumably exists to relate values at different scales. Such a scaling function would model the variation of parameters with scale. A simple example of scaling function is aggregation with a spatial average.

Broadly speaking, there are three ways to find parameter values for a pyramid. They could be measured at different supports, they could be calculated at a single level and then generalised using a scaling function, or they could be calculated at different resolutions from other parameters that have been measured or generalised with a scaling function. The scaling function of a multi-scale model can be stored as a function and solved on demand or calculated at a range of resolutions and stored as multi-scale values in an appropriate data structure. However, as noted in Section §2.3, scaling functions may be unknown and difficult to determine for non-linear processes. Under these circumstances, if multi-scale values cannot be determined by appropriate measurements they may be modelled with an approximation of the true scaling function.

Interpolating a surface from sample points and resampling a surface more sparsely is not the same as representing a value or process at a different scale using a scaling function. The process of fitting a surface and then resampling creates a new point value at a precise location but does not necessarily represent a new scale because it is still a point measurement. However, some scaling functions may involve interpolation.

5.4. Level of Detail Modelling

This section applies the concept of Level of Detail (LOD) surface models to the task of interpreting hierarchical spatial data for distributed geographic models: selecting spatial resolutions and forming adaptive realisations.

5.4.1. Choosing appropriate scale

Occam's razor is a heuristic for choosing simpler explanations that retain the required explanatory power. Cartographic generalisation is choosing simpler representations that

are still effective descriptions. Forming adaptive realisations from a LOD distributed model is choosing coarser scales that continue to represent a process within acceptable limits of accuracy.

For instance, cartographers depict a bridge on a map with a bridge symbol not a collection of symbols for the bridge's parts. This is question of scale and purpose; the cartographer must ensure their symbol is clear enough to tell the map reader that they can cross the river here. A similar philosophy applies to computer modellers of environmental process. It is neither practical nor advantageous to model every detail.

A chosen scale may be too coarse if it disguises sub-pixel heterogeneity that is important to process representation but also too fine for two reasons. Firstly, the resolution may be creating unnecessary computations and secondly, the resolution may be representing information at a resolution for which the true value is unknown and has therefore been interpolated, possibly erroneously.

LOD modelling is a computationally efficient way to produce variable resolution 2.5-D, or 3-D surfaces where the surface resolution varies depending on the distance to the viewer (see Subsection 3.8.3). LOD modelling for distributed geographical models applies a similar principle and methodology. An appropriate scale is selected from multi-scale datasets in three steps.

1. A hierarchical pyramid is formed by a scaling function.
2. An error value is associated with each value in the pyramid.
3. A decision rule is then evaluated against the error value to identify the appropriate scale.

An adaptive resolution realisation is generated from the coarsest cells that satisfy the decision rule (Subsection 5.4.2).

Scaling functions

If the sampling scale or measurement scale of the parameters in a model differ from the model grid scale, interpolation or generalisation are required, which can be accomplished with a scaling function. Caution is required when utilising model elements derived using a scaling function due to a subclass of the fallacy of composition that is well known in statistics.

As noted by Peckham et al. (2009), if X is some model parameter that varies spatially, f is a non-linear function, and $Y = f(X)$ is a computed quantity; then the following

generally holds:

$$E\left[f\left(X\right)\right] \neq f\left(E\left[X\right]\right) \tag{5.4}$$

In the above formula, E is the expected value, which is akin to the spatial average (Peckham et al., 2009). In other words, the result of applying a function to the spatial average of its parameters and variables is not necessarily the same as finding the spatial average of the results of the function evaluated separately from the original distributed values. This is similar to the notion of data support in spatial statistics. Data support is the domain within which linear averages were computed (Dungan, 2001). As the support changes so do the computed values. However, the extent to which Inequality 5.4 holds varies with the nature of the function. In some instances the following holds.

$$E\left[f\left(X\right)\right] \approx f\left(E\left[X\right]\right) \tag{5.5}$$

provided X does not vary too greatly. Due to Tobler's law[3] (Tobler, 1970), if Equation 5.5 holds for limited domains of X, it can be inferred that for sufficiently restricted spatial domains, the fallacy of composition does not apply. This can be exploited to generate simplified LOD realisations.

Error values

In all but entirely uniform surfaces, the generalisation involved in moving from fine to coarse scale entails the loss of sub-pixel variability. The aggregated parent value will diverge from the extreme values of its children. It would be desirable to know the extent to which sub-pixel heterogeneity has been disguised or the extent to which supra-pixel ambiguity has been realised by interpolation for a given resolution, in order to assess the applicability of the generalisation. The extent to which the generalised value disguises the underlying variation can be represented by an error value.

A simple example of an error value is maximum deviation from the value determined by the scaling function. Such an error value compares the parent value to the values of its associated children on the finest level of resolution to give an indication of the quantity of error associated with the realisation compared to the reference layer.

[3]Tobler's First Law of Geography: Everything is related to everything else, but near things are more related than distant things.

Decision rules

The error value can be investigated with a decision rule to assess the appropriateness of the resolution for the purpose to which it is applied. A simple example is to set a threshold which represents the largest acceptable error value. Provided the threshold value is set appropriately, utilising the generalised value in the model will produce results sufficiently similar to applying the model to its parts.

More complex models may require more complex scaling rules, error values, and decision rules. For instance, the decision rule may involve more than one spatially distributed multi-scale variable or the decision rule may include parameters that change spatially in a process analogous to how LOD 3-D visualisation resolution becomes coarser with distance from the observer.

5.4.2. Adaptive realisations

An LOD realisation is a variable resolution surface that has been generated from a multiresolution dataset, typically in real time for visualisation. RHSM adaptive realisation extends the methods of LOD modelling to distributed geographical modelling. Variable resolution modelling involves identifying the coarsest resolution at which a process is represented accurately. RHSM realisations utilise the largest cell in the multi-resolution pyramid at which the error value meets the decision rule.

It could be argued that the proposed RHSM is not multi-scale in the sense that once it has generated the realisation it is a flat surface with a single LoD at every point. However, the underlying operation of the decision rule, threshold value, scaling function and error value are multi-scale and offer a framework to investigate scale effects.

The resolution of adaptive surfaces used in a GIS, such as TINs, are typically defined by the complexity of the underlying surface that is being modelled. The spatial extent of the model element is therefore a function of the variation of the surface. The greater the variation of the surface, the finer the resolution required to represent the surface accurately. Conversely, for a given surface complexity; the greater the cell size, the greater the underlying complexity that is disguised. A variable resolution discretisation of a field in 2-D or 3-D space attempts to minimise sub-pixel heterogeneity whilst using as few cells as possible. RHSM realisations do not necessarily minimise surface variation because the error value may be formulated to capture other aspects of the spatial model.

Not withstanding the variable resolution, an RHSM realisation is a regular structure. Therefore, it is not necessary to encode point locations or tile division processes. The

realisations can be stored on trees similar to quad trees. On a given LoD, each branch may be either a leaf (containing a value) or a node (containing links to child nodes or leaves). The number of nodes between the root to the leaf represents the spatial extent of the value and the labels of the nodes represent its location through a hierarchical referencing system described in detail in Section §6.1.

Unlike irregular triangular LOD models, the proposed model uses regular geometries. The study area is covered without gaps or overlaps by tiles that are similar but not congruent in shape. Variable density realisations formed by RHSM datasets will not have conforming edges. However, variable resolution distributed modelling involves interaction between different resolutions. Therefore, model processes designed to operate on LOD realisations must be structured to be robust to nonconforming edges (see Chapter 8 for hydrological geomorphometry examples).

5.5. Proposed Regular Hierarchical Surface Model (RHSM)

This section draws this chapter to a close by outlining the proposed RHSM.

The criteria of the proposed model are that the surface should:

1. Represent different values of variables at different scales.
2. Provide a way to identify appropriate scale for a given process.
3. Support multi-scale analysis and intra-scale comparisons.
4. Facilitate local neighbourhood operations across multiple scales.
5. Support large datasets, data tiling, and parallel processing.
6. Be applicable to multiple geometries.
7. Minimise computational overheads from delineation of position and scale.

The characteristics of the proposed model are:

1. The RHSM is a functional, multi-scale, regular, implicit model.
2. The geometry of the RHSM has implicit position, shape and hierarchy.
    a) The RHSM employs a regular discretisation of both space and scale.
    b) The RHSM hierarchies are congruent, aligned, and self-similar.
    c) RHSM parent cells have the same geometry as cell neighbourhoods.

d) The RHSM datasets support array rotation and translation.

3. The RHSM supports three regular geometries:
   a) Triangular, RHSM-tri;
   b) Rectangular, RHSM-rec,
   c) Hexagonal, RHSM-hex.
4. The RHSM supports scale variation.
   a) The RHSM has different values for the same area at different resolutions. (pyramids)
      - The RHSM generates pyramids with a scaling function
   b) The RHSM can generate level of detail realisations (variable density).
      - The RHSM generates realisations using an error value and decision rule.
5. The RHSM data model is an array tree.
   a) The RHSM data model consists of array structures as leaves of a tree.
      - The tree/array threshold is user defined.
   b) The RHSM is referenced using an adapted Hierarchical Image Processing (HIP) system.
      - The referencing is common to both array and tree components.
      - Hierarchy is stored implicitly.
      - Hierarchically agglomerated values are adjacent in memory.
   c) Sparse realisations are stored in a tree structure.

This chapter has described a conceptual model a Regular Hierarchical Surface Model that addresses the questions of scale. The next chapter will detail the mathematical techniques required to implement it.

6. Mathematical Regular Hierarchical Surface Model

This chapter describes the mathematics of the RHSM. The indexing method is detailed in Section §6.1, the array-tree data model is presented in Section §6.2 and the procedure for LOD modelling including pyramids, scaling rule, error value, and adaptive realisations is described in Section §6.3. The computational model follows in Chapter 7 and hydrological applications are detailed in Chapter 8.

6.1. Indexing

The RHSM indexes individual tiles using an adaption of the HIP system, the mathematics of the adapted system are described in this section.

6.1.1. The HIP ordinate

Middleton and Sivaswamy (2001) introduced a Hexagonal Image Processing (HIP) system, which included a linear, hierarchical indexing method called the HIP index. The HIP index is only applicable to aperture-7 hexagonal hierarchies. The RHSM described here applies the HIP indexing system to aperture-9 rectangular and aperture-4 triangular hierarchies as well. Therefore in the context of the RHSM, the abbreviation "HIP" is generalised from Hexagonal Image Processing to Hierarchical Image Processing (HIP). The specific form of HIP index is indicated by a superscript number that matches the aperture ($\mathfrak{a}$) of the geometry it relates to. Therefore HIP^7, HIP^9, and HIP^4 are the indexing systems for the hexagonal RHSM-hex, rectangular RHSM-rec, and triangular RHSM-tri respectively.

Some noteworthy characteristics of the HIP index are that

- it expresses 2-D location with a single ordinate,
- the length of the index indicates the number of parts into which the whole has been partitioned,

- the length of the base vector is the resolution of the dataset,
- a self similar hierarchy can be formed by agglomerating the cells that share the same HIP index with the last digit removed, and
- the centre of each HIP cell is the centre of the 0 cell of its children, i.e. it is an aligned hierarchy.

Digits of the HIP ordinate

A HIP ordinate consists of a sequence of numbers that represents the location of a specific value within a HIP dataset. The HIP ordinate indicates both the value's location in memory and the location on Earth that it represents. The cardinality (n) or number of cells in a HIP dataset is fixed by the number of digits (λ) in the ordinate according to:

$$n = \mathfrak{a}^{\lambda} \tag{6.1}$$

Middleton and Sivaswamy (2005b) call λ the level of the HIP dataset. Equation 6.1 places an upper limit on the number of LoDs that are possible within a given RHSM dataset (see Subsection 6.3.1).

Each digit in the HIP ordinate (HIP_i) is restricted to:

$$HIP_i \in 0 \leq \mathbb{Z} \leq \mathfrak{a} - 1 \tag{6.2}$$

Like the Hindu-Arabic numeral system, the individual digits read from left to right from coarsest to finest. However, the individual digits are indexed from finest to coarsest as:

$$HIP_{\lambda-1}...HIP_1HIP_0 \tag{6.3}$$

For example, a 5-level (5λ) HIP^7 ordinate will have 5 digits, $HIP_4HIP_3...HIP_0$ that can take a value from 0 to 6, i.e. 24043.

Determination of location from HIP ordinate

Each digit in a HIP ordinate represents a transformation of a base vector ($\mathbf{C}$). The transformation has two components that are described by distortion matrices: (1) a trans-

formation indicated by the value of the digit that may involve rotation and scaling but is here called a rotation matrix (**A**), and (2) a transformation indicated by the place value of the digit which may also involve rotation and/or scaling but is here called the scaling matrix (**B**).

A, **B**, and **C** vary depending on the geometry and are given below in the descriptions for the different geometries.

C can be scaled to produce the desired cell size. The cell size of an RHSM is defined as the distance between the cell centres of line adjacent neighbours.

The vectors formed by the sequence of transformations contained in a HIP ordinate are summed to give a vector that represents a translation from the origin of the dataset to the location of the value. The location determined from the HIP ordinate is the centre of the tile. The shape of the tile may be assumed from the type of tessellation. The origin is the centre of the central cell. The origin of the HIP dataset is coordinated in a spatial reference system to locate the entire dataset on the surface of the Earth.

The (x, y) vector between the origin of the RHSM dataset and a given value ($\begin{bmatrix} x_\triangle \\ y_\triangle \end{bmatrix}$) can be determined from the HIP ordinate by:

$$\begin{bmatrix} x_\triangle \\ y_\triangle \end{bmatrix} = \sum_{i=0}^{\lambda-1} \mathbf{A}_d \mathbf{B}^i \mathbf{C} \tag{6.4}$$

where i is the index of the HIP digit and d is the value of the HIP digit at index i. If the origin of the RHSM dataset has projected coordinates $\begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$ the location of the value is:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_\triangle \\ y_\triangle \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{6.5}$$

Examples of HIP^7 and HIP^4 location determination are given in Appendix C.1.

Interpretation of distortion matrices

It is useful to note that the distortion matrices can be interpreted conceptually as the unit vectors in the direction of the x and y axes after the required rotation and/or scaling. This interpretation is shown in the equation below and illustrated in Figure 6.1.

$$Unit\ vector\ (1,0) \rightarrow \begin{bmatrix} x_x \\ y_x \end{bmatrix}$$

$$Unit\ vector\ (0,1) \rightarrow \begin{bmatrix} x_y \\ y_y \end{bmatrix}$$

$$Distortion\ matrix \begin{bmatrix} x_x & x_y \\ y_x & y_y \end{bmatrix} \tag{6.6}$$

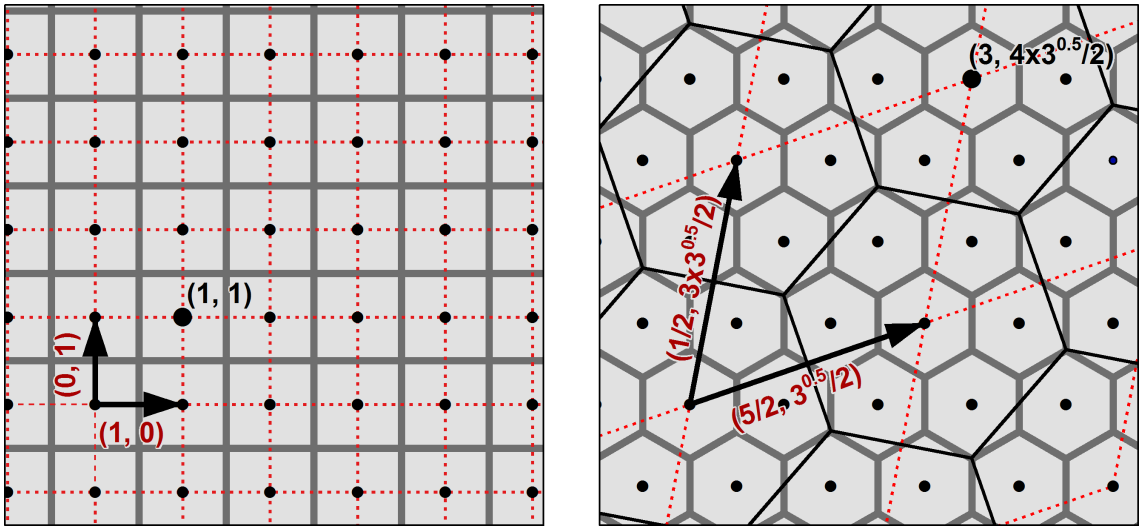


Figure 6.1.: Distortion matrix. The Cartesian unit vectors in (x, y) (black arrows on the left) are converted into the (u, v) unit axes associated with $\lambda 2\ HIP^7$ by tilting the y-axis $\pi/6$ radians clockwise into the hexagonal v-axis, stretching both axis by a factor of $\sqrt{7}$ and rotating both anticlockwise by $\tan\left(\sqrt{3}/2\right)$. The resulting vectors are $Unit\ vector\ (1,0) \rightarrow \begin{bmatrix} 5/2 \\ \sqrt{3}/2 \end{bmatrix}$, $Unit\ vector\ (0,1) \rightarrow \begin{bmatrix} 1/2 \\ 3\sqrt{3}/2 \end{bmatrix}$, therefore we have $Distortion\ matrix \begin{bmatrix} 5/2 & 1/2 \\ \sqrt{3}/2 & 3\sqrt{3}/2 \end{bmatrix}$. The Cartesian coordinates the larger black dot can be converted from the location in the left to that in the right using matrix multiplication:

$$\begin{bmatrix} 5/2 & 1/2 \\ \sqrt{3}/2 & 3\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 4\sqrt{3}/2 \end{bmatrix}$$

HIP arithmetic

HIP indices represent vector quantities because they define the distance and direction from the origin of the dataset to the cell centre. Therefore arithmetic operations can be defined within the HIP ordinate system (Middleton and Sivaswamy, 2005b). HIP addition is represented with the symbol $\oplus$, and subtraction with $\ominus$. Middleton and Sivaswamy (2005b) also define HIP multiplication, however, this is not described here.

In each of the geometry sections below, tables are given to show the full range of possibilities for HIP addition of single digits. Following Middleton and Sivaswamy (2005b), HIP addition can then be calculated similar to ordinary base 10 addition. The HIP addition tables show the result of adding the digit in the first column to the digit in the first row; where the result is two digits, the first digit is carried.

Subtraction can also be performed by determining the value that represents the opposite vector, which is the negation, and then performing the addition of the negation. Negation tables are also given for each geometry below. The neighbourhood of a cell in an RHSM dataset can be calculated by addition (Middleton and Sivaswamy, 2001). Examples of HIP arithmetic are given in Appendix C.2.

6.1.2. Hexagonal

The RHSM-hex uses the HIP^7 referencing system, which is essentially equivalent to the HIP referencing system described in Middleton and Sivaswamy (2001). HIP^7 forms aligned aperture-7 hierarchies.

The value of each digit within a HIP^7 ordinate ranges from 0 to 6 and represents the rotation of the base vector by multiples of $\pi/3$ radians anticlockwise from the x-axis, except 0, which defines the centre of the agglomeration. Rotation matrices ($\mathbf{A}$) for HIP^7 are given in Table 6.1 along with the scaling matrix ($\mathbf{B}$) and base vector ($\mathbf{C}$). These matrices are not unique, for instance Middleton and Sivaswamy (2001) use a slightly different formulation. Part of a 3λ RHSM-hex tessellation is shown in Figure 6.2.

HIP^7 hierarchies are not congruent because the 7 hexagon agglomerations are not strictly self-similar. Rather than being hexagons, the agglomerations have a complex shape that tends toward a fractal as the number of agglomerated cells increases. However the agglomerations can be simplified into hexagonal forms by determining the Voronoi diagram of the centroids of the agglomerations. Hexagonal tessellations formed by this method rotate between levels by $\tan^{-1}\sqrt{3}/2$ (see dashed lines in Figure 6.2). The fractal shapes are further discussed in Subsection 7.3.5

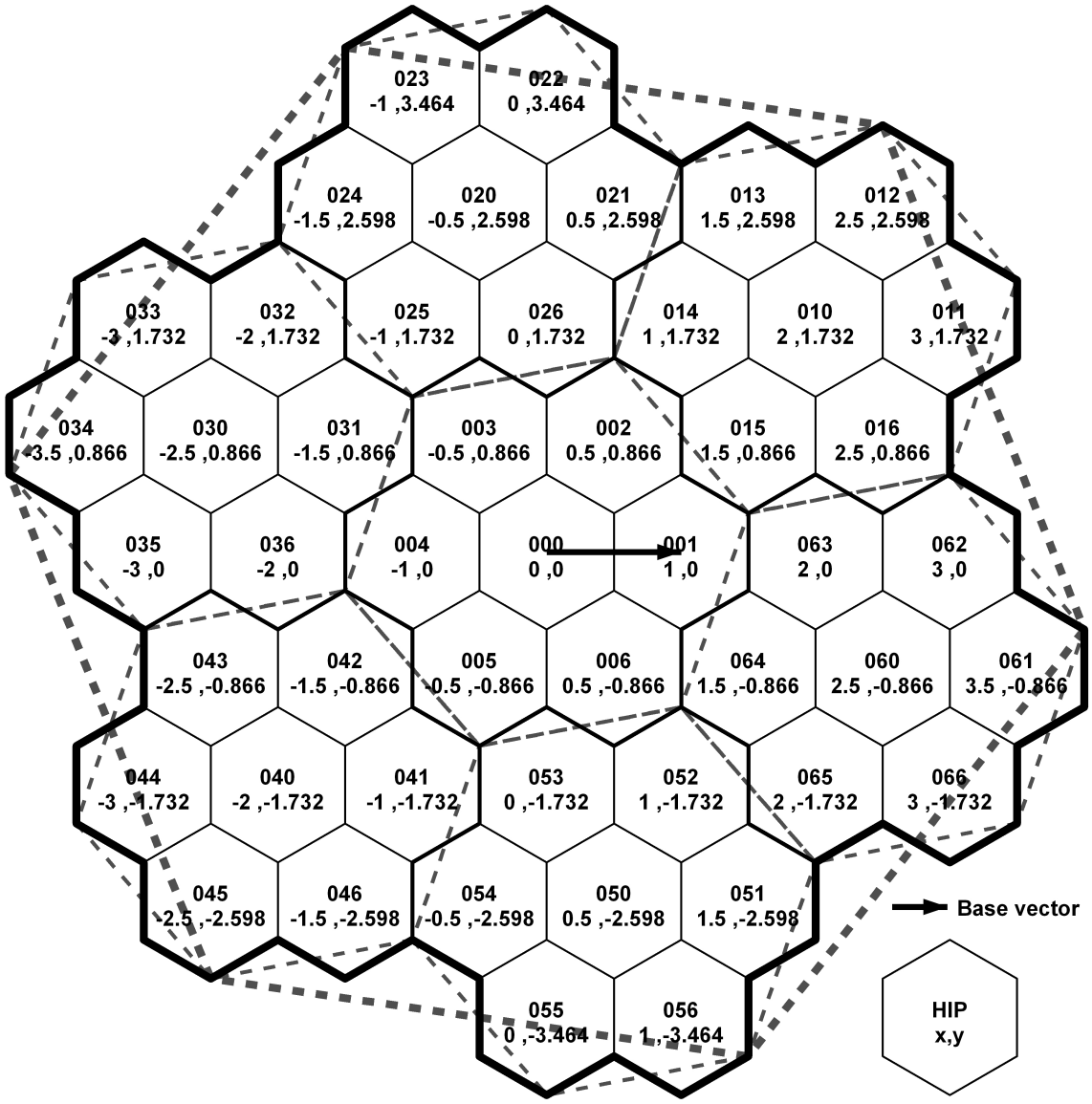


Figure 6.2.: The HIP^7 referencing system for part of a 3λ RHSM-hex partition of space. Hexagons are grouped into aggregates of 7 for each coarser LoD. Voronoi tessellations are shown for all three levels (dashed lines). Reproduced from Wright et al. (2014)

6.1.3. Rectangular

HIP^9 was developed for the RHSM-rec based on an aligned, congruent, aperture-9 hierarchy (Figure 6.3). RHSM-rec hierarchies are self-similar. **A**, **B**, and **C** for HIP^9 are given in Table 6.2. In the RHSM-rec the scaling matrix is in fact a scalar. For the line adjacent neighbours (HIP^9 values 1, 3, 5, and 7) the rotation matrix only rotates, for the point adjacent neighbours (2, 4, 6, and 8), the rotation matrix scales and rotates. Part of a 3λ RHSM-rec tessellation is shown in Figure 6.3.

Table 6.1.: HIP^7 parameters. cs is the cell size of the HIP^7 dataset.

(a) HIP^7 conversion to (x, y)

HIP digit (d)	Rotation ($\mathbf{A}_d$)
0	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
1	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
2	$\begin{bmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{bmatrix}$
3	$\begin{bmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix}$
4	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
5	$\begin{bmatrix} -1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & -1/2 \end{bmatrix}$
6	$\begin{bmatrix} 1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & 1/2 \end{bmatrix}$

Scaling ($\mathbf{B}$)	Base vector ($\mathbf{C}$)
$\begin{bmatrix} 2 & -\sqrt{3} \\ \sqrt{3} & 2 \end{bmatrix}$	$\begin{bmatrix} cs \\ 0 \end{bmatrix}$

(b) HIP^7 Addition table ($\oplus$).

	0	1	2	3	4	5	6
0	0	1	2	3	4	5	6
1	1	63	15	2	0	6	64
2	2	15	14	26	3	0	1
3	3	2	26	25	31	4	0
4	4	0	3	31	36	42	5
5	5	6	0	4	42	41	53
6	6	64	1	0	5	53	52

(c) HIP^7 Negation table ($\ominus$).

0	1	2	3	4	5	6
0	4	5	6	1	2	3

Table 6.2.: HIP^9 parameters. cs is the cell size of the HIP^9 dataset.

(a) HIP^9 conversion to (x, y)

HIP digit (d)	Rotation ($\mathbf{A}_d$)
0	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
1	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
2	$\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$
3	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
4	$\begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix}$
5	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
6	$\begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}$
7	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
8	$\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$

Scaling ($\mathbf{B}$)	Base vector ($\mathbf{C}$)
3	$\begin{bmatrix} cs \\ 0 \end{bmatrix}$

(b) HIP^9addition table ($\oplus$).

	0	1	2	3	4	5	6	7	8
0	0	1	2	3	4	5	6	7	8
1	1	15	14	2	3	0	7	8	16
2	2	14	26	38	37	3	0	1	51
3	3	2	38	37	36	4	5	0	1
4	4	3	37	36	48	52	51	5	0
5	5	0	3	4	52	51	58	6	7
6	6	7	0	5	51	58	62	74	73
7	7	8	1	0	5	6	74	73	72
8	8	16	15	1	0	7	73	72	84

(c) HIP^9negation table ($\ominus$).

0	1	2	3	4	5	6	7	8
0	5	6	7	8	1	2	3	4

044 -4,4	043 -3,4	042 -2,4	034 -1,4	033 0,4	032 1,4	024 2,4	023 3,4	022 4,4
045 -4,3	040 -3,3	041 -2,3	035 -1,3	030 0,3	031 1,3	025 2,3	020 3,3	021 4,3
046 -4,2	047 -3,2	048 -2,2	036 -1,2	037 0,2	038 1,2	026 2,2	027 3,2	028 4,2
054 -4,1	053 -3,1	052 -2,1	004 -1,1	003 0,1	002 1,1	014 2,1	013 3,1	012 4,1
055 -4,0	050 -3,0	051 -2,0	005 -1,0	000 0,0	001 1,0	015 2,0	010 3,0	011 4,0
056 -4,-1	057 -3,-1	058 -2,-1	006 -1,-1	007 0,-1	008 1,-1	016 2,-1	017 3,-1	018 4,-1
064 -4,-2	063 -3,-2	062 -2,-2	074 -1,-2	073 0,-2	072 1,-2	084 2,-2	083 3,-2	082 4,-2
065 -4,-3	060 -3,-3	061 -2,-3	075 -1,-3	070 0,-3	071 1,-3	085 2,-3	080 3,-3	081 4,-3
066 -4,-4	067 -3,-4	068 -2,-4	076 -1,-4	077 0,-4	078 1,-4	086 2,-4	087 3,-4	088 4,-4

Figure 6.3.: The HIP^9 referencing system for RHSM-rec. The top line is the HIP^9 ordinate and the bottom line is the Cartesian (x, y) coordinates. The origin is the centre of the central rectangle in both systems. Two levels of a 3λ dataset are shown. Reproduced with corrections from Wright et al. (2014).

6.1.4. Triangular

HIP^4 was developed for the RHSM-tri based on an aligned, congruent, aperture-4 hierarchies (see Figure 6.4). RHSM-tri hierarchies are self similar. **A**, **B**, and **C** are given in Table 6.2.

The base vector in HIP^4 differs from HIP^7 and HIP^9 due to the alternating orientation of the cells in the tessellation. Within a single LoD of an RHSM-tri there are two different orientations for both cells and their neighbourhoods. The central cell in each

agglomeration points in the opposite direction to its neighbours. In contrast, rectangular cells do not vary in orientation and hexagonal tessellations are consistent on a single LoD but rotate between levels.

RHSM-tri neighbourhoods consist of a central cell and its three line adjacent neighbours. The vectors that describe the translation from a cell to its neighbours differ depending on the orientation of the central cell. To describe the tessellation using only four rotation matrices it is necessary to first identify whether λ is odd or even and then change the sign of the base vector for every zero index to the left (coarser) of the index being considered (see Table 6.3).

Table 6.3.: HIP^4 parameters. cs is the cell size of the HIP^9 dataset.

(a) HIP^4 conversion to (x, y)

HIP digit (d)	Rotation ($\mathbf{A}_d$)
0	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$
1	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
2	$\begin{bmatrix} -\sqrt{3}/2 & 1/2 \\ -1/2 & -\sqrt{3}/2 \end{bmatrix}$
3	$\begin{bmatrix} \sqrt{3}/2 & 1/2 \\ -1/2 & \sqrt{3}/2 \end{bmatrix}$

Scaling ($\mathbf{B}$)	Base vector ($\mathbf{C}$)
$\begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$	$\begin{cases} if \left(\lambda - \sum_{HIP_i}^{HIP_\lambda} 1\left[HIP_i = 0\right]\right) is\ even & \begin{bmatrix} -cs \\ 0 \end{bmatrix} \\ if \left(\lambda - \sum_{HIP_i}^{HIP_\lambda} 1\left[HIP_i = 0\right]\right) is\ odd & \begin{bmatrix} cs \\ 0 \end{bmatrix} \end{cases}$

(b) HIP^4 addition table ($\oplus$). HIP^4 addition is not commutative as indicated by the asymmetry of the addition table. The table is read $row \oplus column$ i.e.$2 \oplus 1 = 13$.

	0	1	2	3
0	0	1	2	3
1	1	0	23	32
2	2	13	0	31
3	3	12	21	0

(c) HIP^4 negation table ($\ominus$). Due to the direction flip, HIP^4 digits are their own negation.

0	1	2	3
0	1	2	3

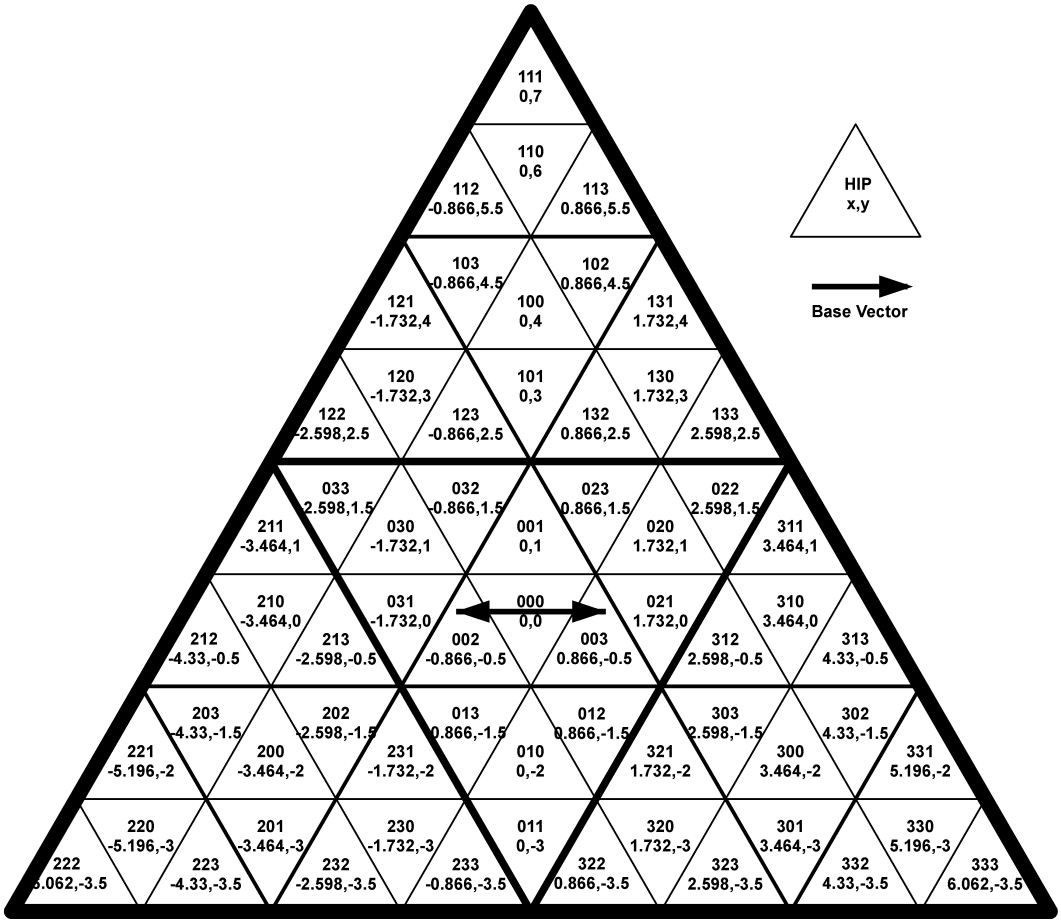


Figure 6.4.: The HIP^4 referencing system for the RHSM-tri. The top line is the HIP index and the bottom line is the Cartesian (x, y) coordinates.The origin is the centre of the central triangle in both systems. Three levels are shown. The base vector is aligned with the x-axis. The base vector of the central "0" cells changes sign for each level.

6.2. The array tree

The HIP ordinate provides a convenient and intrinsically hierarchical way of indexing spatial data. But what data model should be used? Two options were considered: a linear array or a tree structure. It is also possible to use an explicit key-value database but that would require unnecessary overheads and a separate DBM system, besides, the HIP index lends itself to intrinsic storage.

The selected structure combines tree and array data storage. The RHSM decomposes the HIP dataset into consistently indexed tree and array components. Separated by an aggregation value, the coarser elements of the HIP ordinate form the tree structure, the finer elements form the array structure. However, the array component is not linear but instead multi-dimensional. The array tree structure is described here from a mathematical perspective, for the computational implementation see Section §7.2.

6.2.1. The tree

The coarsest digits of the HIP ordinate are used as labels for the nodes of a tree structure. If values are stored on the leaves of the tree, the HIP index can be determined by tracing the branches back to the root, and, if required, the HIP index can be converted into an (x, y) location using Equation 6.4. An example of how location is encoded by the tree branches is shown in Figure 6.5. The initial branch is marked L0 to indicate that this is the base level to distinguish it from other LoDs in a pyramid (see Subsection 6.3.1).

This storage method is very similar to a quad-tree except that for RHSM-tri the four branches divide space into triangular geometry, and for hexagonal and rectangular geometries there are seven and nine branches respectively. The tree data structure can be sparse or have variable density, however, if a regular tree is used where each node has $\mathfrak{a}$ children, a regular structure is generated that reflects the RHSM hierarchy aperture.

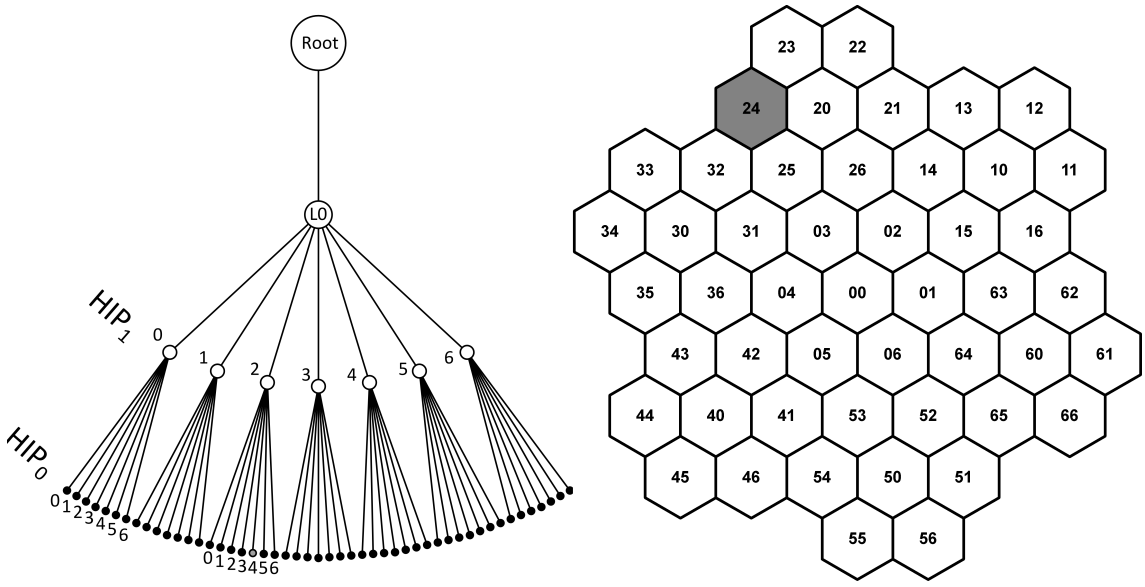


Figure 6.5.: The tree component of the HIP^7 referencing system. The grey node corresponds to the grey hexagon and has $2\lambda HIP^7$ reference 24.

6.2.2. The array

The RHSM does not store individual values on the tree, instead the RHSM stores arrays[1] attached as leaves to a tree data structure. The array tree is similar to the tree and tile division used by Platings and Day (2004) and the Bing Maps Tile System (Microsoft Corporation, 2013). However, those systems use the hierarchical index only for the tree;

[1]In this description an array is a table of fixed dimension that contains homogeneous data, typically floats or integers.

the arrays are conventional images with implicit row, column coordinates; whereas, the RHSM utilises the hierarchical indexing method in the array component as well.

The HIP index is implemented in the array by treating the HIP index as the coordinates of a value in an array with λ dimensions and length $\mathfrak{a}$ in each dimension. Therefore, a $\lambda HIP^{\mathfrak{a}}$ array has dimension $\mathfrak{a}^{\lambda}$. For instance, a value with a 3λ HIP^7 index of 043 would be stored as a value at (0, 4, 3) in an array with dimension 7^3. This example is shown in Figure 6.6. There is a minimum of one array in each RHSM. Large datasets can be broken into multiple arrays.

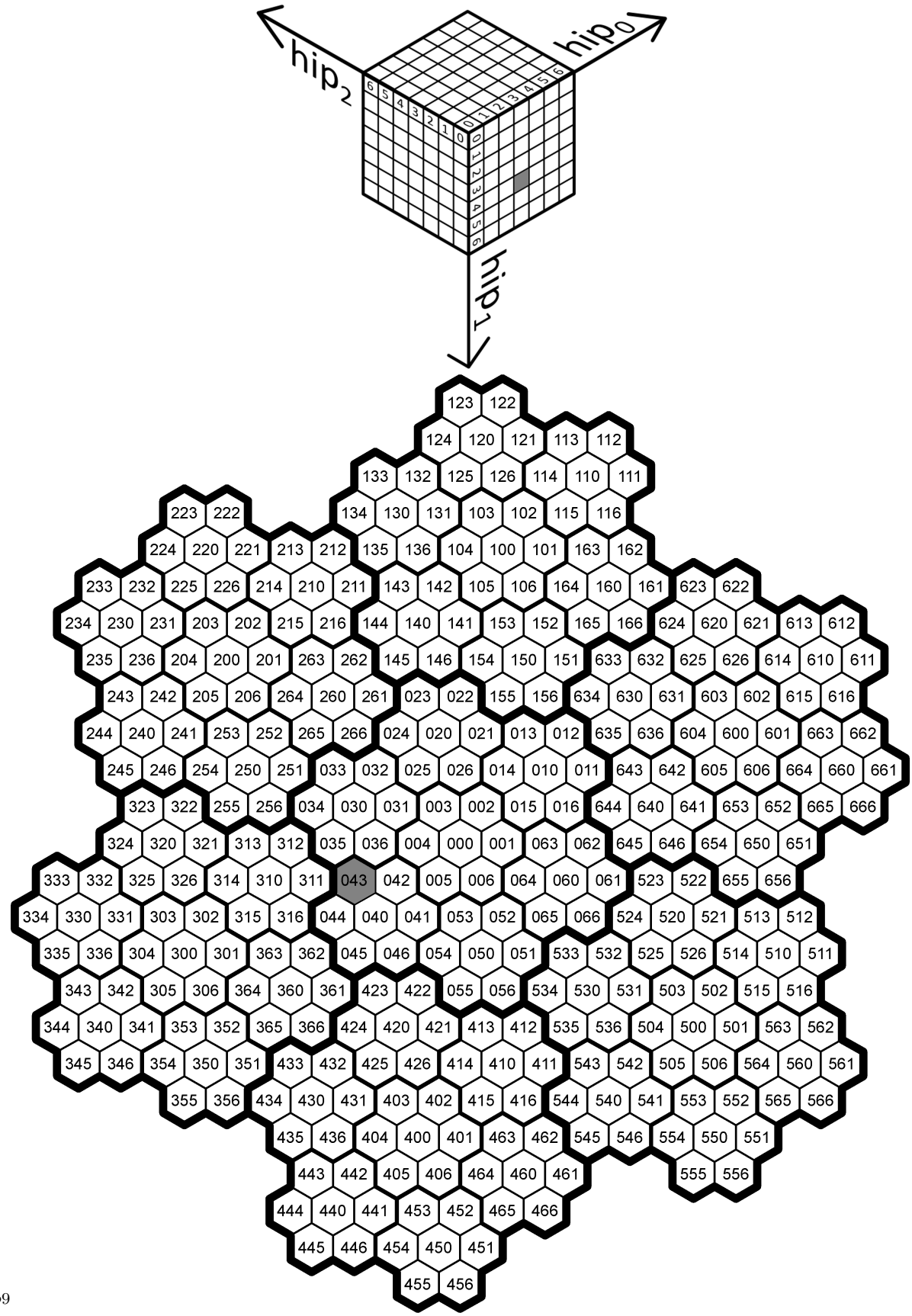
Utilising the multidimensional array ensures that the same referencing system is used throughout the dataset and that the hierarchical structure is implicit in both the tree and array components. This is useful not just for spatial analysis that relies on the hierarchical nature of the data, such as variable resolution flow direction arrays, but also when combining datasets and dividing arrays into new branches (Wright et al., 2014). In addition, it is very simple to select the set of values that are children of a given node or to find summary statistics of the finest children to create a smaller, coarser array. This is useful for forming pyramids (see Subsection 6.3.1).

6.2.3. The aggregation value

A user defined aggregation value (α) separates the tree and array and components of a HIP ordinate in the RHSM. An n level RHSM dataset, with an aggregation value of x and aperture of $\mathfrak{a}$ can be specified as $n\lambda^{\mathfrak{a}}x\alpha$.

The aggregation value sets the maximum size for the array component in an RHSM. The digits of HIP ordinate whose indices are larger (coarser) or equal to the aggregation value are stored as node labels in the tree component. Those that are finer than the aggregation value form the coordinates of an array. Choosing an appropriate aggregation value is a trade-off between using few large tiles that entails reading unnecessary data or many small tiles that create more reads. Excessively small or large tiles adversely affect performance.

A schematic of an RHSM-hex data structure with five levels of detail and an aggregation value of three is shown in Figure 6.7. The RHSM-rec data structure is similar except that there are nine branches on each node of the tree, and nine values in each dimension of the array. In the RHSM-tri, there are four branches on each node, and four values in each dimension.



HIP^9

Figure 6.6.: Array component of the HIP^7 referencing system. The grey square corresponds to the grey hexagon and has $3\lambda HIP^7$ index 043.

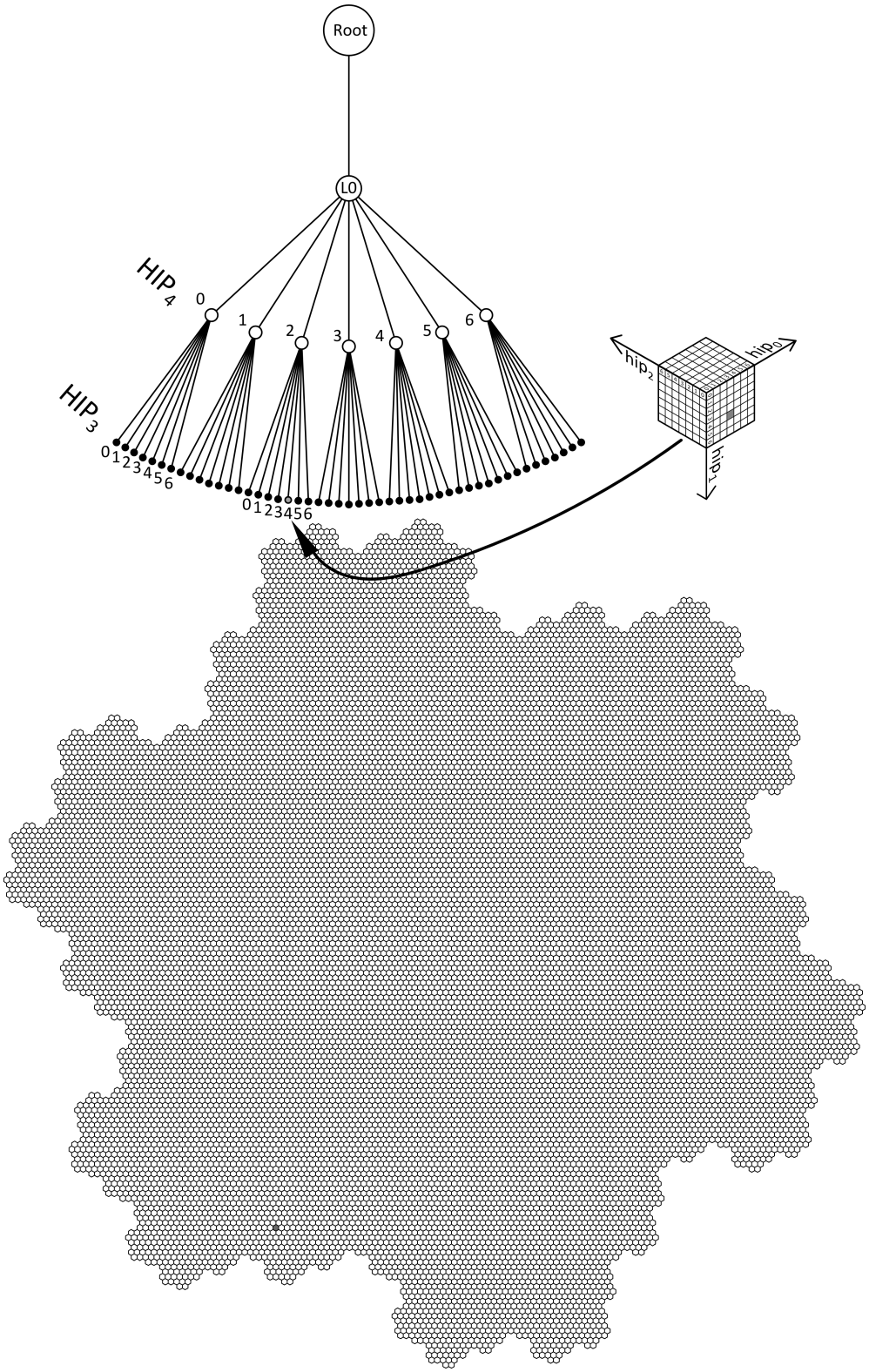


Figure 6.7.: A five level RHSM-hex data structure with an aggregation value of three ($5\lambda^7 3\alpha$). The grey square matches the grey hexagon and has HIP ordinate 24043. The rotation between levels causes the discrepancy of location of the grey tile with Figure 6.5.

6.3. Level of Detail modelling

This section discusses the mathematical aspects of Level of Detail modelling in the RHSM: pyramid calculation, error values, and adaptive realisations.

6.3.1. Pyramid calculation and referencing

Pyramids are a series of representations of a given region at successively coarser resolutions. Pyramid layers are a form of generalisation. They are often used to facilitate rapid visualisation of raster datasets with changing viewer position or zoom. However, within the RHSM they are used to store different scale datasets for modelling.

A pyramid dataset requires values to store. In general terms, these data are created by a scaling rule. Some simple examples of a scaling rule are the summary statistics: arithmetic mean, median, or mode. For some processes more complex scaling functions may be required. Summary statistics are an approximation, therefore, an error value is also required, which can be investigated to ensure the suitability of using the generalised value for modelling (see Subsection 6.3.2).

A coarser LoD RHSM pyramid layer consists of a new array with $1/\mathfrak{a}$ as many cells. A λ level dataset can generate up to $\lambda + 1$ pyramid layers including the base layer and the coarsest layer, which consists of a single cell (see Figure 6.8, which shows the pyramid layers of a 5-level RHSM-hex data structure, with an aggregation value of three). From finest to coarsest the LoDs in an RHSM pyramid are labelled

$$Lx\{0 \leqslant x < \lambda, x \in \mathbb{Z}\} \tag{6.7}$$

The RHSM treats all LoD of the pyramid as separate branches of a network tree at root level. However, it would also be possible to place pyramid values on nodes of the base LoD because the length of the HIP ordinates would indicate the LoD resolution.

In all LoD except the base level, each cell has $\mathfrak{a}$ child cells. The parent-child relationship is expressed implicitly by the HIP reference. The coarser parent pixel's HIP ordinate is the same as the beginning of its children in the finer levels of detail, making it a trivial task to relate cells in different LoDs to their descendent and ascendants. For example, the $L2$ in a $5\lambda^7$ with the HIP^7 ordinate of 240 has the set of finer children: {2400, 2401, 2402, 2403, 2404, 2405, 2406}.

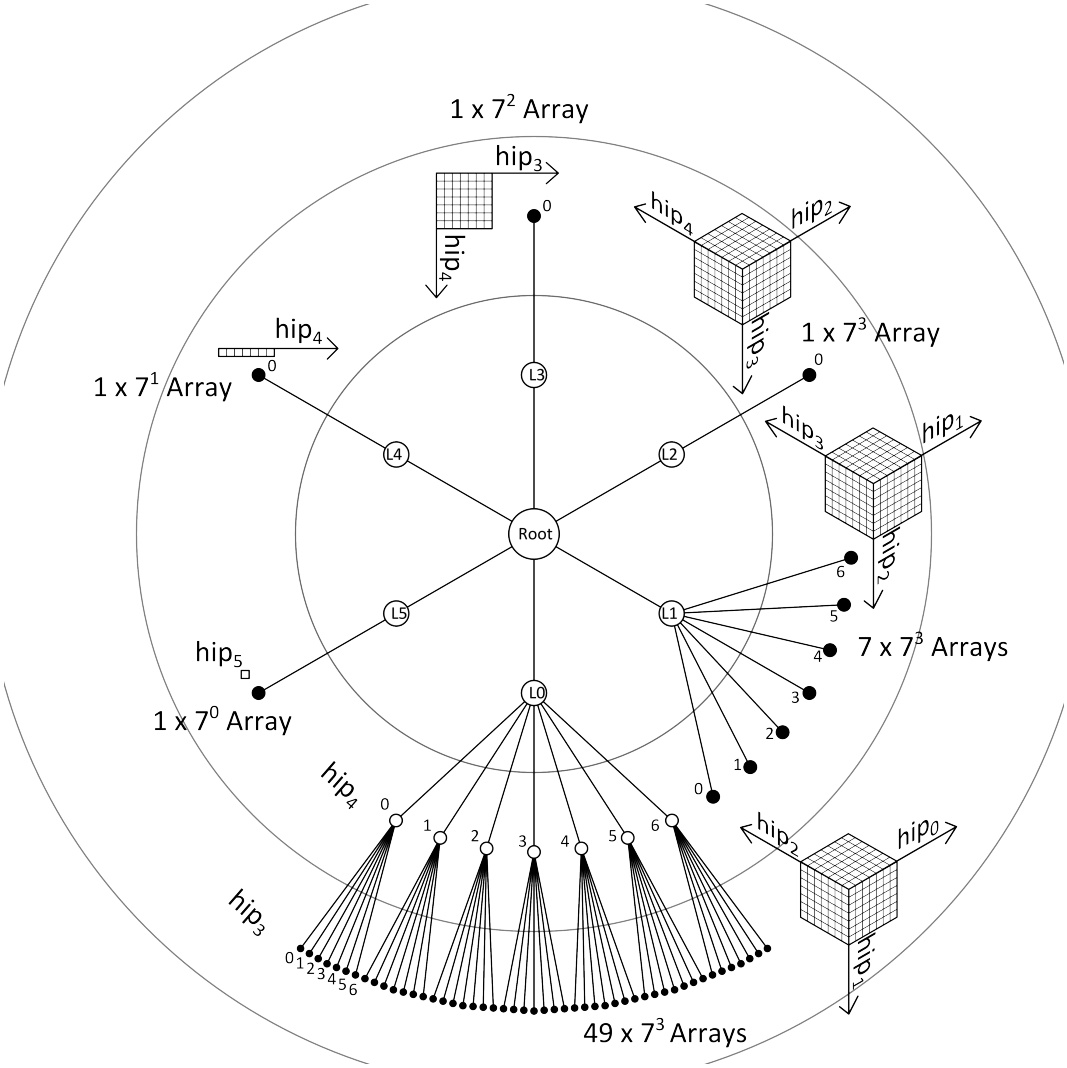


Figure 6.8.: A five level RHSM-hex data structure with pyramids and an aggregation value of three ($5\lambda^7 3\alpha$).

The aggregation value is enforced in the pyramid levels, therefore the LoDs are divided into array-trees to ensure that the individual arrays are no larger than the aggregation value. The aggregation level should be set to reflect memory limitations. Arrays on coarser LoD pyramid layers have the maximum number of dimensions allowed by the aggregation value. Enforcing the aggregation value in the pyramid levels in this way proved to be problematic because processing operations are computationally complicated by the presence of coarser arrays that bridge multiple finer arrays. An alternative approach, that may be more effective, is to allow the dimensionality of the array to reduce until they are only single values and then combine them to form a single array at the level corresponding to the aggregation value: effectively defining the maximum tree size instead of the maximum array size.

6.3.2. Error values

The error value is an indication of the difference between a generalised value and the underlying non-generalised values. Error values are evaluated recursively with a decision rule to assess how suitable a given LoD is for variable density realisation (see Subsection 6.3.3). Error values in the RHSM are stored in array-trees very similar to the array-trees that store the values. The only difference is that the branch at the root level is labelled Ex to distinguish it from the values, which are labelled Lx (see Figure 6.9).

A simple example of an error value, which is implemented in Chapter 9, is maximum deviation of a value from its descendent. In the following equation, $L0_{HIP}$ is an array of size $\mathfrak{a}^i$. All members of $L0_{HIP}$ are evaluated to determine the maximum variation. The error value is compared to $L0$ not $L\,(i-1)$ because $L\,(i-1)$ is already generalised.

$$Ei_{HIP} = \max\left(\left|Li_{HIP} - L0_{HIP,j,k\ldots}\right| \{j,k\ldots \in 0 \leq \mathbb{Z} \leq \mathfrak{a},\}\right) \tag{6.8}$$

Equation 6.8 implies that $L0$ values have no errors and are, therefore, the true scale of the data. However, fractal surfaces do not have a true scale; all scales of fractal surfaces are approximations. Alternatively, error values could be determined from variograms or fractal dimension. However, such approaches have not been considered in this study.

6.3.3. Adaptive realisation

An adaptive realisation is a variable density surface extracted from a multi-resolution, hierarchical dataset where the element size in the realisation is adapted to ensure the error value conforms to a decision rule. Adaptive realisations in LOD models are usually intended for real-time visualisation of large datasets. In the RHSM they are generated for geographic modelling.

Decision rule

The adaptive realisation is generated by a recursive process. Starting at the coarsest resolution, the error values are assessed by evaluating a decision rule $D\,(E)$. If the decision rule evaluates to True, the resolution is acceptable and the value is assigned to the realisation. If the decision rule evaluates to False the children are assessed. The process continues until all branches have been assessed as True or the base resolution has been

reached, in which case the base resolution is used. Alternatively, if the base resolution has non-zero error, No Data could be returned.

A simple example of a decision rule that is used in Chapter 9 is to introduce a threshold value (T). If the error value is below the threshold value the decision rule evaluates to true in accordance with:

$$D\left(Ei_{HIP}\right) = \begin{cases} Ei_{HIP} \leq T & True \\ Ei_{HIP} > T & False \end{cases} \tag{6.9}$$

The decision rule can include parameters that vary spatially; for instance, in LOD models for visualisation, it is typical for the threshold value to increase with distance from the observer. An example from geomorphometric hydrological analysis is that the threshold could decrease with increasing flow accumulation.

Sparse tree

The adaptive realisation is stored in a sparse tree structure. There is no array component of the sparse tree because arrays imply a regular distribution. The adaptive realisation tree is labelled S on a branch at root level. There is only one S branch at any given time and generating a new one will overwrite the existing one. There are no LoD in the S branch. An RHSM-hex dataset including error values and a sparse tree is shown in Figure 6.9.

Each node in the sparse tree contains either $\mathfrak{a}$ branches or is a leaf value. The HIP ordinate of a sparse value determines the labels of the nodes it is attached to in the same way as the Li and Ei trees determine the coarse element of HIP ordinates. It is the values in the sparse tree that are used for variable density hydrological modelling in Chapter 9.

The areal extent of each cell in the sparse tree (A_s) can be determined from the length of its HIP ordinate (h) as defined by its node labels as:

$$A_s = A_b \mathfrak{a}^{\lambda - h} \tag{6.10}$$

where A_b is the area of the cells in the base layer.

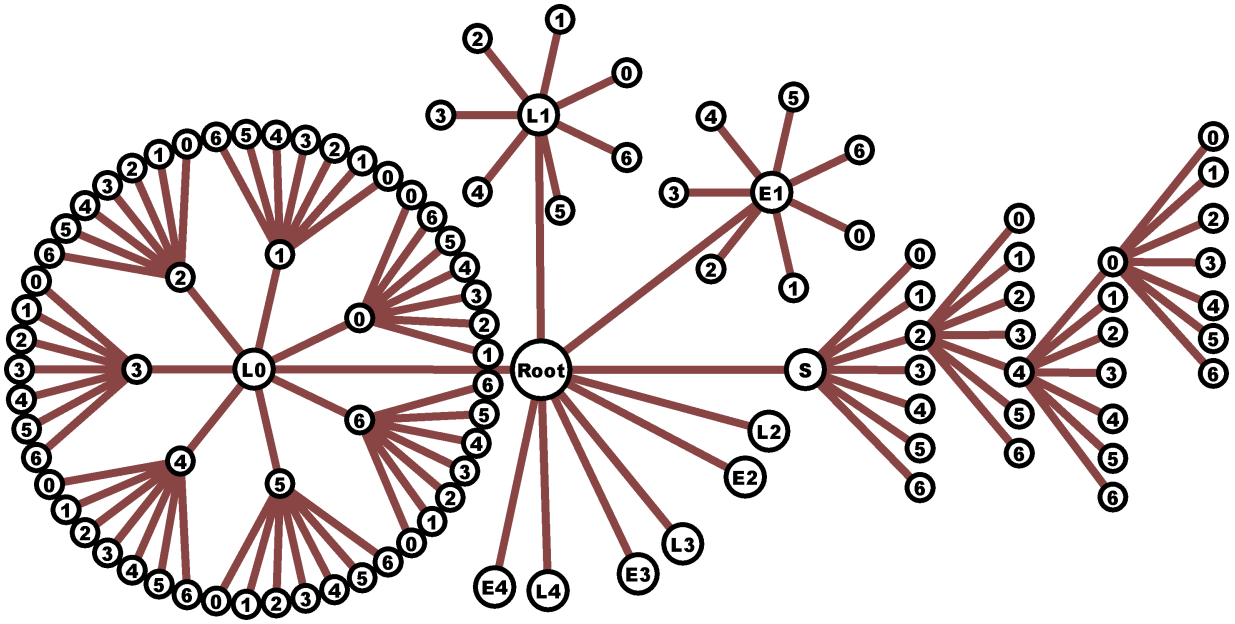


Figure 6.9.: Array tree including error values and a sparse arrays. The tree structure shows a RHSM-hex $5\lambda^7$.

This chapter has detailed the mathematical tools required to implement the surface model described conceptually in Chapter 5; it has described the indexing method, the array tree data model, and the Level of Detail techniques required to generate pyramids and variable density realisations. The next chapter details how these concepts were realised in the computational model.

7. Computational Regular Hierarchical Surface Model

This chapter describes the computational model of the RHSM. Section §7.1 gives an overview of the components of the RHSM computational model and the remaining sections describe those components in more detail: Section §7.2 describes the implementation of the datafile, Section §7.3 describes how the RHSM was integrated with a GIS, and Section §7.4 details the Python object class.

Implementation and tests were conducted on a machine with an AMD Athlon(tm) II X4 640 Processor @ 3 GHz , 4.00 Gb RAM, with SATA 6.0 Gb/s 3.5″disk storage @ 7200 RPM, using Windows 7 Professional 64-bit operating system with an NTFS file system. This computer is henceforth described as a "commodity computer." In this chapter RAM is termed "main memory", and hard drive is termed "secondary memory".

7.1. RHSM Overview

There are four components of the RHSM computational model, the user, the datafile, the GIS representation of the data, and an object class encoded in Python that connects the other three parts. The datafile stores the spatially distributed parameter as NumPy arrays in a hdf5 (hierarchical data format version-5) file, which can be accessed in a Python script using the PyTables package. The RHSM was integrated with the commercial GIS ArcGIS. The data are displayed in the GIS as ArcGIS feature classes, either polygon or point. The GIS representation allows data visualisation and interaction with other data inside a GIS. The Python object stores metadata, manages the relationship between the datafile and the GIS feature classes, and provides methods for the user to interact with the data. The overall structure is shown in Figure 7.1.

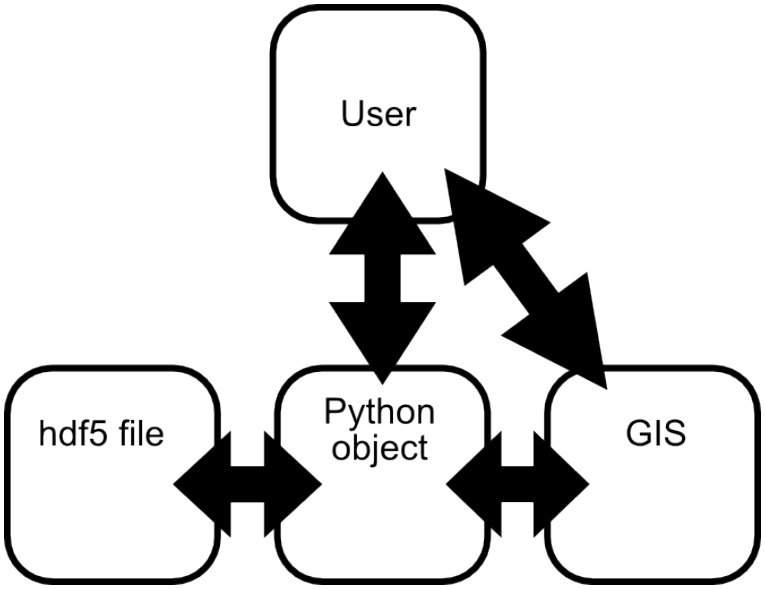


Figure 7.1.: Overview of the components of the RHSM. The black arrows indicate interaction.

7.2. Hierarchical data file

Various options were considered for storing and accessing the data. Several considerations were important in choosing the final data model and scripting module for the implementation. The hierarchical structure needed to be efficiently implemented so that the data can be rapidly accessible and searchable (Subsections 7.2.1 and 7.2.1), and large datasets needed to be supported (Subsection 7.2.3).

7.2.1. PyTables and NumPy

The RHSM mathematical model defines the data model structure to consist of a tree and an array component. This array tree structure is compatible with hdf5 file structure, which organises arrays in a hierarchical tree. The Python module PyTables was chosen to implement the RHSM. The implementation uses PyTables to manipulate a hdf5 tree with NumPy arrays hanging off the nodes. The entire tree and the associated arrays are a single file.

Python was chosen due to ease of development and cohesion with the scientific code biosphere that has developed around Python. Python is optimised for easy writing and rapid development, not for fast computation. Scripts will execute quicker if processing is performed using the tools provided by PyTables and NumPy, where possible, as these modules wrap around C libraries that execute more efficiently than Python. Subsection 10.4.1 suggests ways to further improve RHSM performance.

The Python module PyTables (Alted et al., 2002) is a multi-purpose software module designed to manage hierarchical datasets and efficiently cope with extremely large amounts of data. This format combines the computational abilities of the NumPy module and PyTables' efficient algorithms for memory optimisation, search speed and I/O efficiency, including automatic balancing of nodes in main memory.

The RHSM is able to generate pyramid layers very quickly because the values of each agglomeration at the finest LoD in a λ-level RHSM are stored adjacent in dimension λ of the array. NumPy tools are designed to efficiently perform calculations on the λth dimension of an array. This can be exploited to quickly summarize values in each agglomeration along dimension λ to create a new smaller array, which can be summarised the same way.

It is possible to only save the base level ($L0$) and determine the pyramid values on the fly. Such a system needs more processing time and may need to repeat calculations but also ensures that the data are consistent and uses less memory. However, the RHSM calculates the pyramid and stores the values on disk. This saves time if the pyramid layer is used repeatedly but risks data conflicts. Using the predetermined pyramid levels, if $L0$ is altered, the user must then recalculate the pyramid. The hierarchical structure can be used to restrict the recalculation to only affected areas.

7.2.2. Representing HIP ordinates in Python scripts

The RHSM stores position implicitly. However, even though the HIP ordinate is not stored explicitly, during processing it is sometimes necessary to explicitly represent a HIP ordinate. The Python tuple data type was chosen for this purpose.

There are a number of built in data-types that could have been used to represent HIP ordinates in Python including numerical types, integer or float; and compound types such as string, tuple, list, or dictionary. Integer would be the least memory intensive. However, integers and floats cannot readily be indexed (you cannot directly investigate what the ith value is). Lists and dictionaries were excluded because they are unhashable, therefore, they cannot be used as keys in python dictionaries. Although the core data model does not use python dictionaries, some processing tools may benefit from organising parts of an RHSM in a dictionary with the HIP ordinate as key. Strings are hashable but would require type conversions to be used in calculations. Therefore, tuple is the most appropriate built-in Python data type.

The tuple data-type can be used to index NumPy arrays directly and matches the conventional way of writing Cartesian coordinates. However, PyTables does not support tuple

names for nodes or as paths expressions, therefore, conversions are required to convert HIP ordinates expressed as tuples into a PyTables address format, which takes a where, node, index form.

7.2.3. Memory management

The RHSM can support very large datasets. PyTables uses the C library hdf5 for secondary and NumPy for main memory processing to efficiently manage arrays that are much larger than those that can be stored in a system's main memory. The data are stored in secondary memory and individual arrays are transferred into main memory during processing[1]. The task of managing the nodes in memory is undertaken by PyTables, which retains the most recently used nodes. The array component must be small enough to be stored in main memory. To allow for operations that involve multiple arrays it is necessary to accommodate several arrays in main memory. Therefore, to prevent memory errors, the maximum array dimension should be reduced by 1λ.

The tree component of an RHSM dataset is generated using an intermediate array. This places an upper limit on the tree size, which is similar to the array size (although somewhat higher as the tree array need not be float). Therefore, the RHSM can theoretically support datasets with cardinality up to $\mathfrak{a}^{2(m-1)}$ where m is the number of dimensions of the largest array that can be stored in main memory and $\mathfrak{a}$ is the aperture of the dataset. Although the exact size of the array that can be used will depend on available computer power, Figure 7.2 indicates that the theoretical maximum is unrealistically large for a commodity computer. Generating array-trees becomes very slow for large trees but is also slow for large arrays. Peak performance occurs around an aggregation value of 4 for small RHSM datasets and then increasing with λ after that.

For a commodity computer running 4GB RAM, RHSM-hex can theoretically support up to $16\lambda 8\alpha$ datasets. This equates to a HIP^7 structure consisting of 7^{16} (approx 33 trillion) hexagons, which would require 260 TB of Hard Drive space to store in an hdf5 file (this is less space than required to store the NumPy array because PyTables compresses the data). Such file sizes are not readily accommodated by commodity computers.

A manageable but still large file size of 100 GB would comfortably contain a $\lambda 11 HIP^7$ dataset. However, datasets of this cardinality would be impractically slow to process on a commodity computer. Therefore, processing speed is the practical restriction on

[1]PyTables is safer for write-once/read-many applications and can only read or write one process at a time. Alted et al. (2002) advise to use locking to ensure data security. Working with secondary memory, risks corrupting it. Therefore, backing up is essential. Working in main memory, storage is effectively backed up.

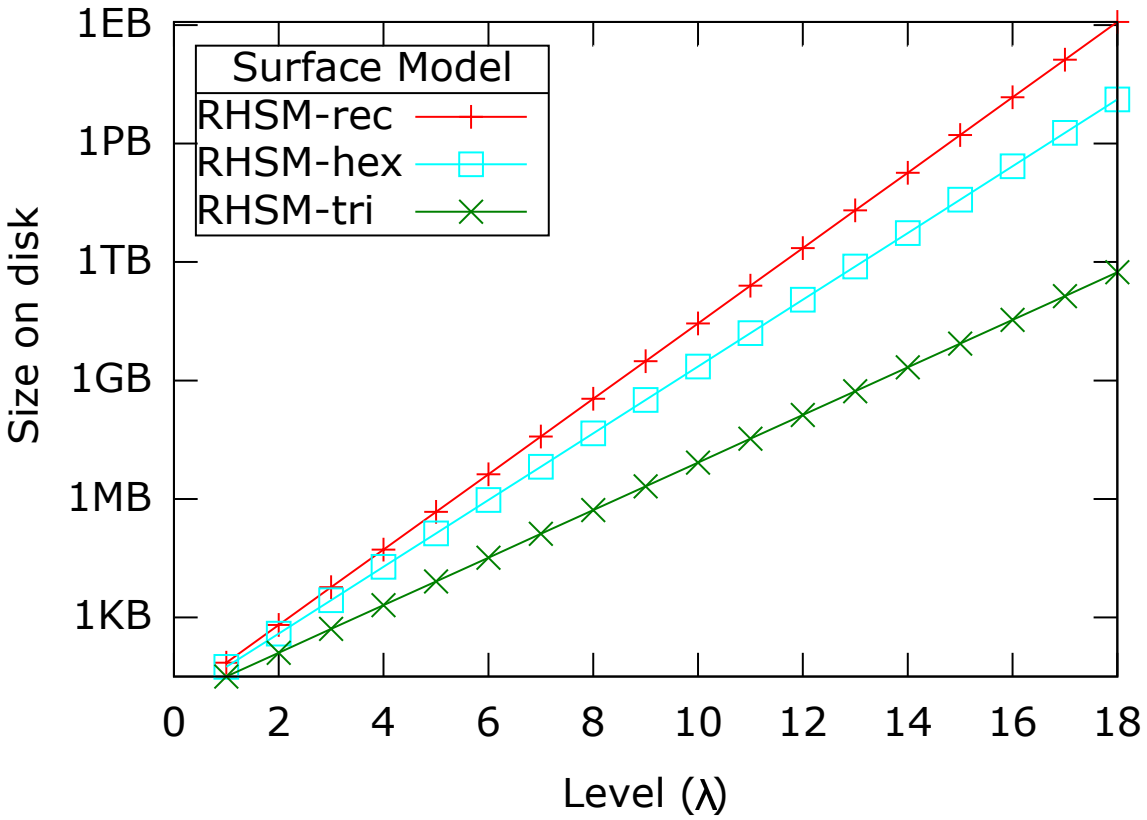


Figure 7.2.: Size of RHSM datasets on disk. The size represents the size in memory of a 64bit float NumPy array. For a given λ, a lower $\mathfrak{a}$ will result in fewer array elements and less memory requirements.

RHSM dataset size not main memory. Figure 7.3 indicates the extent and λ of RHSM datasets required to cover central Dunedin, New Zealand with 0.1 metre cell size, which, is a resolution common in ortho-imagery and achievable in DEMs generated by LiDAR or drone based photogrammetry.

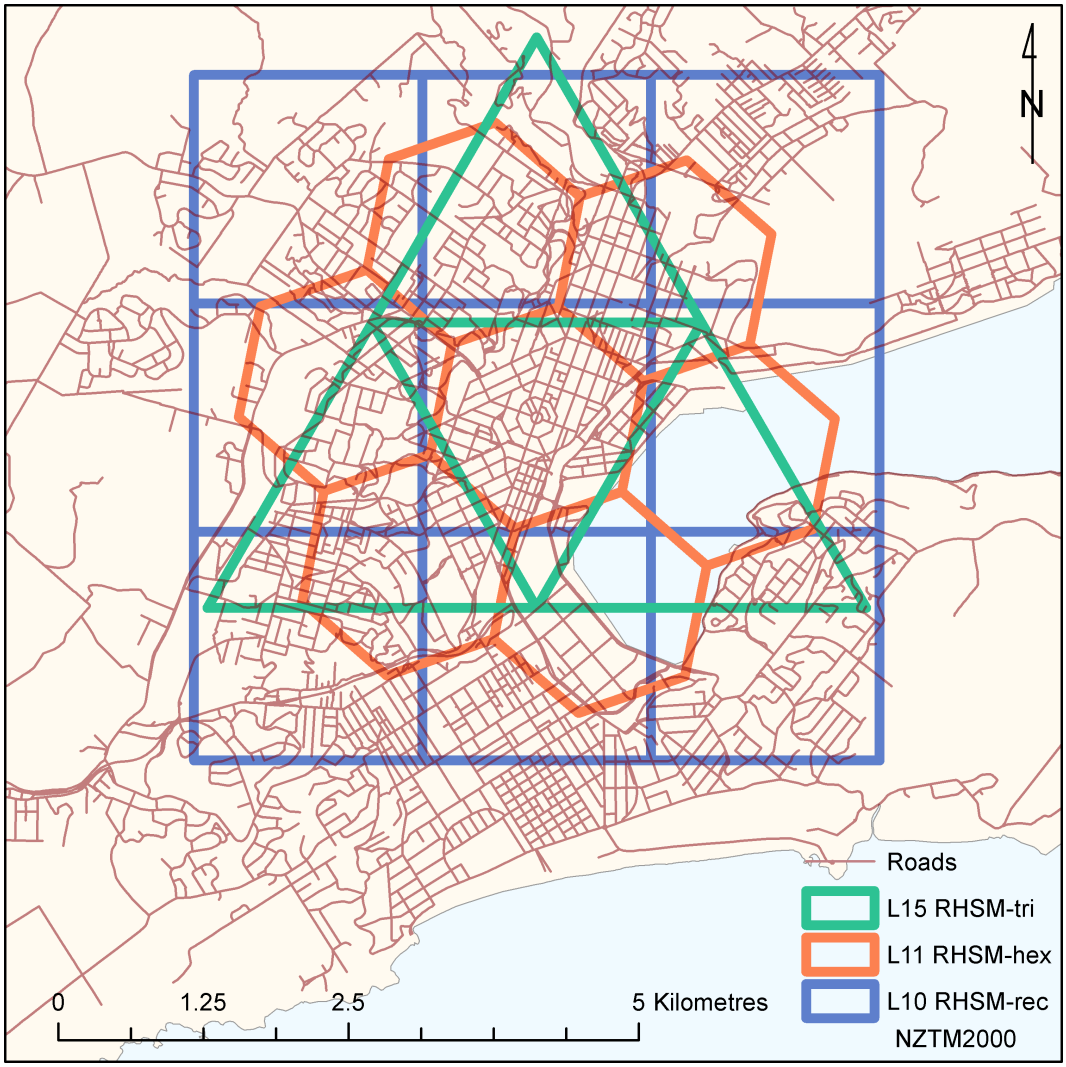


Figure 7.3.: Extent of RHSM datasets covered by comparatively large (but still less than 100GB) RHSM datasets. L0 cells have a cell size of 0.1 metres (not shown). $\lambda - 1$ shown. The extent of the datasets is compared to the Dunedin urban area, a city of approximately 100 000 people. A dataset one level larger would be required to completely cover the entire Dunedin urban area.

7.3. GIS integration

Given that the RHSM does not comply with GIS data model standards, existing general purpose GIS packages are not able to interact directly with RHSM data without custom coding. Therefore, the RHSM is integrated in a GIS using a three tier structure, the data model in hdf5, the RHSM object as a Python class, and the GIS representation as an ArcGIS feature class.

The GIS used was ArcGIS 10.1. ArcGIS is a proprietary software package with closed source code, which limits possible functionality. ArcGIS was chosen for three reasons: (1) to demonstrate the RHSM within the constrictions of a conventional GIS, (2) to develop the software in an application with a large pool of potential users, and (3) to access the diverse range of tools available in ArcGIS.

This section describes how the RHSM was integrated with ArcGIS. Subsection 7.3.1 explains the GIS data model employed, Subsection 7.3.2 explains how RHSM representations are converted to ArcGIS feature classes, Subsection 7.3.3 examines the graphical techniques used to represent the RHSM in a GIS, Subsection 7.3.4 explains how data are updated and deleted, Subsection 7.3.5 considers how the complex shapes created by hexagonal agglomerations should be represented, Subsection 7.3.6 details the interpolations that were implemented to convert GIS data into RHSM format.

7.3.1. GIS data model

RHSM datasets are represented in ArcGIS using feature classes. Feature classes are vector datasets supporting polygon, line, and point structures. RHSM datasets can be represented in ArcGIS as a tessellation of polygon features or as a set of point features. Representing RHSM datasets in GIS using vector layers facilitates vector analysis operations, such as overlay and select by attributes. Feature classes store attribute data in an attribute table. The attribute table of an RHSM feature class contains the HIP ordinate with each index in a separate field. This can be used to select cells or agglomerations of cells according to HIP ordinates using select by attributes.

Feature classes nominally have very large size limits (1TB by default), but they are restricted to 4,294,967,295 features, which is still a very large number. However, using a commodity computer, feature classes struggle at much smaller sizes. For example, feature classes with many features are very slow to redraw at far zoom, and their attribute tables lag, do not show all values, and do not show the correct values. These problems only affect the display; tools like select by attributes work correctly. This performance issue occurs

for datasets significantly smaller than the 4GB main memory limit of the commodity computer. For instance, RHSM-hex struggles at $\lambda 7$ (823,543 hexagons); whereas, $\lambda 6$ works well (117,649 hexagons). It is, therefore, necessary to limit the size of RHSM datasets displayed as feature classes. It is practical and convenient to utilise the aggregation value from the RHSM to limit the feature class size.

A single RHSM can create numerous feature classes. For example, if $\alpha < \lambda$, multiple feature classes may be required for each LoD, for several pyramid levels, an equivalent number of error levels, and a sparse realisation. It will not always be necessary to draw all the feature classes. The RHSM module allows the user to choose which LoD to draw, in part or in entirety; or the user may choose to draw only the sparse feature class. Only one sparse realisation is generated at any given time. If a new one is generated it will overwrite the existing sparse feature class. If the user wishes to retain sparse feature classes, the user must copy the sparse layer to a different location.

To more easily manage the collections of feature classes that represent an RHSM dataset, the feature classes are included in a Feature Dataset, which also ensures that a common spatial reference is maintained. The purpose of each feature class is indicated by a naming convention: each feature class is named according to the RHSM filename appended by an underscore followed by a letter to indicate type (L=data values, E= error values, and S= sparse) and a number to indicate the LoD[2].

Another option would have been to use a raster data model for the GIS representations. Using the raster data model would lessen memory restraints because rasters are already optimised for large datasets and rapid redrawing using pyramids. Within the ESRI software suite, Tile Packages could be developed to allow user control of the layer development. However, only the RHSM-rec would be easily represented as raster. The other geometries could only be represented using finer resolution raster images. In addition, fine resolution rasters would not recognise the actual model elements making tasks like selecting elements with a mouse difficult. Polygon selection is more intuitive because whole polygons are selected.

The above GIS representations assume a tessellated structure for the GIS representation of the RHSM, in accordance with the conceptual model. However, if the dual network was used for the GIS representation, the following options could be utilised:

- Line feature classes could represent RHSM as networks.
- TINs could represent RHSM-rec and RHSM-hex as networks but not RHSM-tri networks, which are hexagonal.

[2]The implementation then appends a series of 0s that also indicate level but these are largely redundant and can be ignored.

- Terrain structures, which are a multi resolution TIN dataset implemented in ArcGIS, could be used to represent the multi-scale structure.

Further suggestions for for development of the GIS integration are described in Subsection 10.4.1.

7.3.2. Drawing feature classes

RHSM datasets consist of several classes of data: a base layer, a series of pyramids, associated error layers, and a sparse dataset. The RHSM "draws" feature classes by determining the geometry from the HIP ordinate and then defining the geometry and other fields line-by-line using the insert cursor. This method does not take full advantage of the implicit nature of the data and is very slow for large feature classes with a large number of vertices.

Drawing RHSM feature classes using the insert cursor is very inefficient because the geometries could readily be defined for the whole feature class due to the implicit regular point set. Unfortunately, ArcGIS 10.1 does not offer a means to assign multiple rows of geometry simultaneously. An alternative method involves translating pre-existing templates. Unfortunately there are no scripting tools that can translate vector datasets in ArcGIS 10.1[3]. That leaves two options, utilising projection tools to adjust the data or using an update cursor to move the location of points in the geometry. Resorting to the update cursor to translate data is unlikely to be significantly faster than drawing the data.

The projection method involves taking an appropriately sized template in a default projection, creating a custom projection to represent the required translations and rotations, running the projection tool to move the data and then resetting the projection to the original projection leaving the data in the adjusted position. Translations can be achieved by adjusting the false origin, and scaling and rotation can be achieved by adjusting transformation parameters. The re-project method was developed but, although marginally quicker for large datasets, it was not as fast as anticipated. However, this approach may be useful in the future if different tools are made available or if the RHSM is implemented in a different GIS.

[3]Shift, rotate and rescale functions are available for raster data (Toolboxes\System Toolboxes\Data Management Tools.tbx\Projections and Transformations\Raster) and the equivalent transformations can be performed using the spatial adjustments tool-bar. The adjust function in the spatial adjustments tool-bar can also be called from a saved text file of transformation vectors. However, there appears to be no way to call these functions from a script.

7.3.3. Visualisation: symbology and scale ranges

GIS representations of RHSM data consist of either point or polygon feature classes. Point feature classes can be symbolised as points, or appropriately sized and rotated point symbols to create a pseudo-tessellation point feature classes. Point feature classes could be generated with pyramid values included in the attributes of the base layer features and symbolised with different sized symbols depending an the pyramid layer required, circumventing the need for multiple data sets.

RHSM feature classes represent their parameter values with colour. If the dataset is separated into multiple feature classes, the colour range needs to be stretched across the entire dataset. The colour scheme is implemented with a representation layer. ArcGIS representations permanently store symbology options such as colour so they can be applied to multiple datasets. The template consists of 32 colours + 1 transparent for no-data, which creates an effect similar to a stretched choropleth.

There are frustrating redraw pauses when viewing fine resolution vector data from far zoom. This can be partially mitigated by not displaying the finest resolution data or by only drawing the sparse dataset. In addition, visible scale ranges can be applied if multiple pyramid layers are present so that fine resolution data are not displayed when zoomed out.

7.3.4. Deleting data / Transferring / Updating data

A weakness of the three tier structure is that it risks data conflicts because data are duplicated. There is data in the form of feature classes and data in the form of an hdf5 file. The hdf5 file can be stored anywhere, however, it is currently implemented to be stored in the same Geodatabase as the Feature Dataset, with the same name.

The dangers of duplicate data are two-fold. Firstly, edits in the feature class will not automatically be reflected in the hdf5 file and, secondly, data may be orphaned if feature classes or the hdf5 files are deleted. This could be prevented by storing the data only in a feature class and converting to RHSM for processing. However, memory constraints make this infeasible, which is why the hdf5 data layer is used as the primary data layer.

One solution to the orphan data problem is to require the user to delete the hdf5 file in addition to the feature dataset. Alternatively, a delete RHSM tool could be provided that deletes all associated files. Both of these options depart from the normal work-flow and may not be adhered to, potentially resulting in junk files accumulating in geodatabases.

7.3.5. RHSM-hex: Voronoi or fractal

The geometry of the underlying data in an RHSM-hex dataset consists of fractal hexagonal rosettes that are formed by dissolving hexagons together (see Figure 7.4). These fractal shapes can be simplified into rotated hexagons by finding the Voronoi tessellation of the centre of each cell at a given level of detail. GIS representations of RHSM-hex datasets are visualised as hexagons at all LoDs rather than the fractal shapes. The symbol used to represent RHSM-hex tiles is not important; for example, a circle could be used instead. The hexagonal simplification is used because it represents the array structure more clearly by identifying its neighbours via a line boundary. A downside of using the hexagonal simplification is that sparse realisations are not tessellations: both gaps and overlaps occur.

The Voronoi simplification of hexagon-like fractal shapes misidentifies the location of some cells within the RHSM hierarchy. In datasets with more than 3 levels, the centres of some cells are located within the hexagonal simplification of the neighbouring super cell rather than the one to which it is assigned by the HIP^7 referencing system and to which its value is contributing. The number of cells misallocated in this way increases with λ. In a 4λ dataset 96 cells are misallocated out of 2401 or 4%, 5λ : 1440/16,807 or 8.5%, : 6λ 15,672 /117,649 or 13%. This creates a discrepancy between the visualisation and the underlying data, however, coarser resolutions are only used where the underlying geography is largely homogenous so it is unlikely that the misidentification would substantively affect modelling results.

Fractal rosettes are not good for GIS representations because they have very complex shapes for coarser LoDs. Rotated hexagons have only six vertices; whereas, the fractal agglomeration has 6×3^n vertices, where n is the LoD ($n = 0$ being the reference level). For coarse representation, the number of vertices in each polygon becomes impractical for visualisation. The memory restraints from the number of vertices are similar to the restraints from the number of records, which makes the fractal rosettes too complex to display.

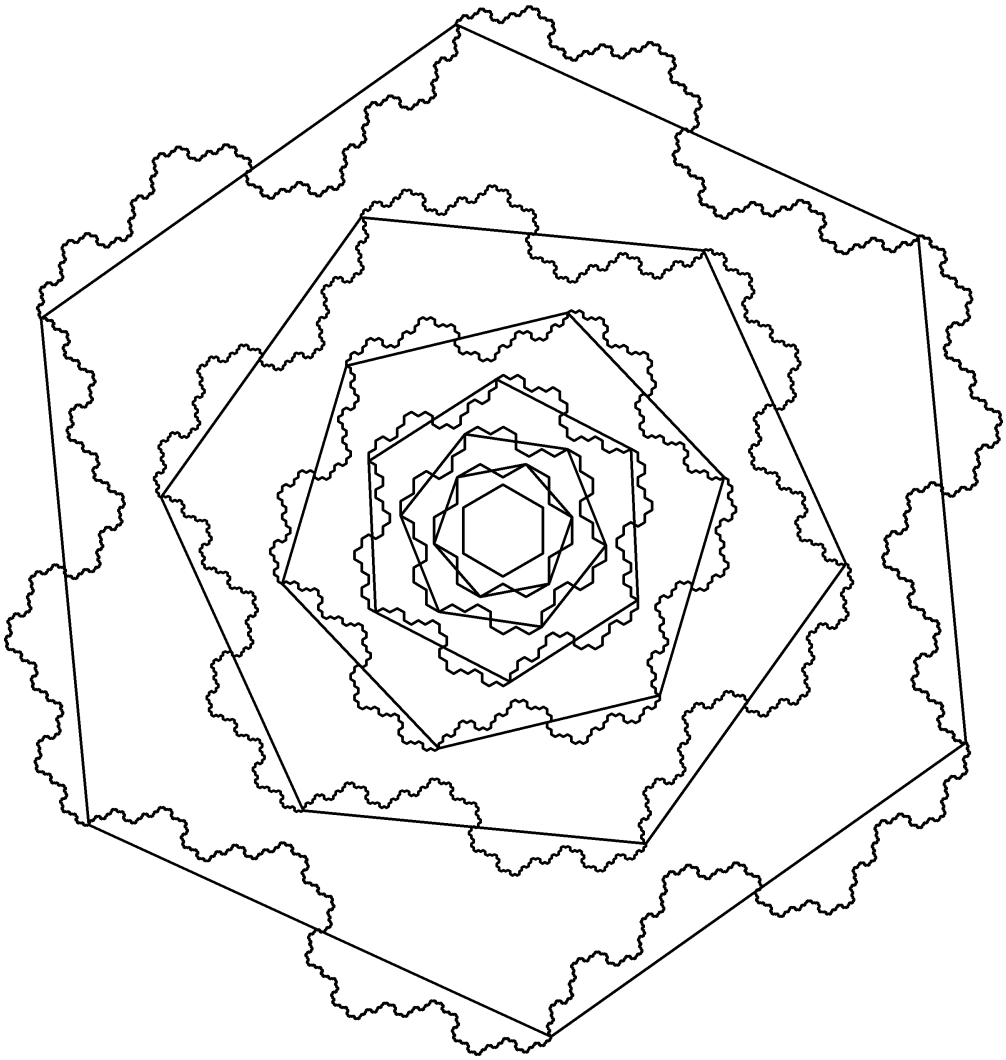


Figure 7.4.: Fractal hexagonal rosettes for an $\lambda 6$ RHSM-hex with Voronoi patterns also shown. Not to scale: finer cells have been enlarged.

7.3.6. GIS to RHSM Interpolation

This subsection describes how data are transferred from GIS datasets to RHSM datasets. GIS data will typically be available in vector, raster, TIN, or contour form. These data are the source data that needs to be resampled to the target RHSM geometries by interpolation.

The simplest interpolation is to assign to the RHSM tile the value of the source DEM or TIN at the tile's centroid. However, unless the source data are reasonably smooth and at a finer resolution than the target geometry, centroid based interpolation creates significant artefacts. These artefacts can be mitigated by considering a larger area in the interpolation; for instance by averaging the values from the tile's corners in addition to the centre. It is necessary to weight the corners according to the number of cells they contribute to.

The analysis in Chapter 9 is largely restricted to mathematically defined surfaces where exact values can be defined at point locations; therefore, interpolation is not required. Where real surfaces are analysed, simple interpolation methods are employed. More accurate interpolation methods should be developed to advance the work into real world environments.

For accurate work, two methods are appealing: Kriging, because it provides statistical information on the likely errors of the interpolation; and the Condat method because it is reversible. The Condat method can only be employed for regular arrays where the density of points is the same between the target and source datasets, which includes rotation of rasters and rectangular to hexagonal conversions. Another option is first or second order least squares spline interpolation. For hydrological applications interpolations should consider drainage structure.

The following methods were implemented in the RHSM:

- centre sampling from square raster and tin,
- corner sampling from square raster,
- areal projection (first order spline) from square raster,
- Condat method from square raster to RHSM-hex,
- and synthetic values from equations.

Interpolation directly from point, or contour layers was not implemented; an intermediate raster or TIN are required.

Real world data are not usually shaped in one of the RHSM geometry agglomeration patterns. Therefore, a No Data value is required. Like the raster data structure, RHSM

datasets have a pre-defined shape and the cells that are outside the study area still exist on disk. Such cells need to be classified as No Data cells and ignored in calculations. The RHSM assigns a value to No Data, this is typically the lowest possible value but it could take a different value.

7.4. RHSM class specification

In order to implement the RHSM, three core objectives were identified. These objectives were:

1. To develop an RHSM python object class that,
    a) contains the variable field being modelled in the form of a HIP indexed array-tree in hdf5,
    b) contains any other attributes required including cell-size, number of levels of detail, origin and file name,
    c) provides tools to create pyramids and variable density realisations, and
    d) provides methods to analyse the data in an RHSM file, including hydrological analysis.
2. To develop Python scripts to efficiently generate an RHSM dataset from given input GIS data: point, raster, or TIN.
3. To develop Python scripts to produce polygon or point feature class representations of RHSM.

These objectives were achieved using the structure and methodology described below. The hydrological modelling tools are described separately in Section §8.6. The description below is not exhaustive, some technical details are skipped for brevity, and segments of code that were used in development but later made redundant are also omitted. The full code is available for download on GitHub (Wright, 2017).

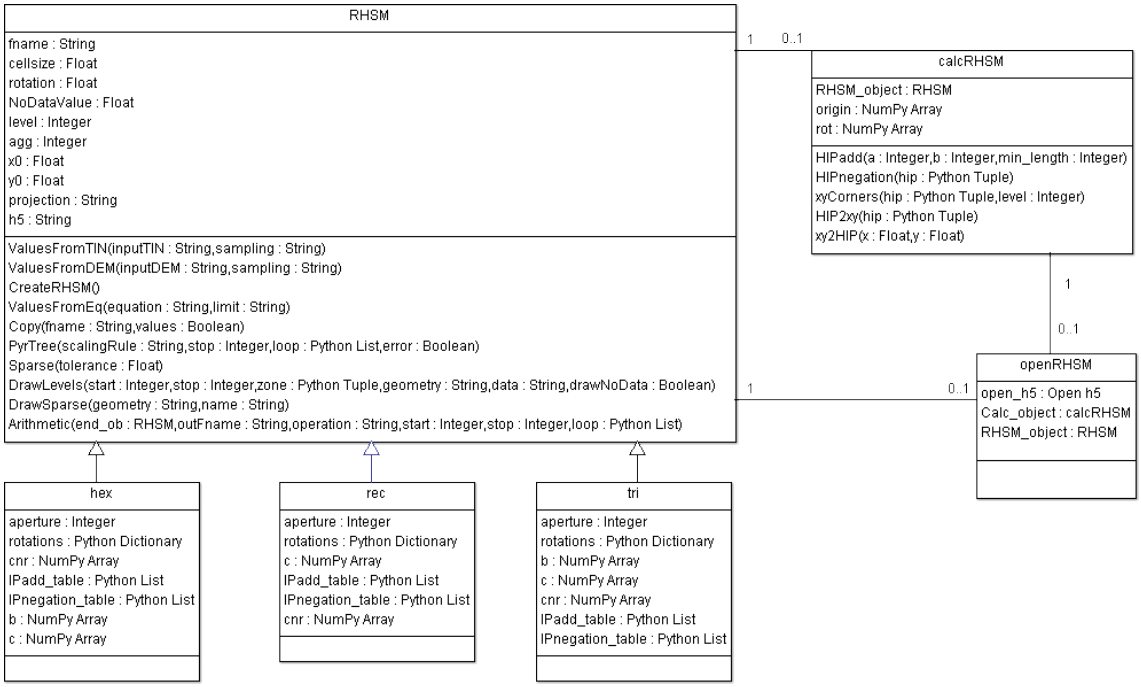
7.4.1. RHSM class structure

The Python class structure is depicted using Unified Modelling language (UML) convention in Table 7.1. The computational model was implemented using Python 2.7, PyTables 2.4.0 and NumPy 1.8.2 in accordance with the mathematical model described in Chapter 6.

The RHSM superclass stores the attributes required to interpret the h5 file such as level and aggregation (agg) values; and those that place it in space such as projection, origin (x0, y0), cellsize, and rotation. The subclasses hex, rec, and tri store class variables that are geometry specific, such as the variables required for indexing (Section §6.1), and drawing polygons (cnr).

Two other classes are encapsulated for specific purposes. Both take an RHSM object as a variable. One, calcRHSM, performs calculations specific to individual cells within an RHSM, the other, openRHSM stores an open hdf5 file. They are encapsulated to prevent repeatedly generating arrays and reopening datasets for common tasks.

Table 7.1.: RHSM UML Class Specification



7.4.2. RHSM class use

Methods are provided in the main RHSM to manage the data. CreateRHSM generates an hdf5 file in accordance with the agg and level attributes, which is initially set to NoData. The RHSM can then be given values using ValuesFromTIN, ValuesFromDEM, or ValuesFromEq; PyrTree, generates the pyramid layers, and Sparse generates the variable resolution realisation. The RHSM is converted to an ESRI GIS representation using DrawLevels or DrawSparse. Finally copy produces a copy of an RHSM dataset and arithmetic performs a function similar to a raster calculator. Any cell by cell calculations implemented in NumPy can be performed with the arithmetic method.

This chapter has described the computational model of the RHSM in general terms in Section §7.1 and as a detailed UML diagram in Section §7.4. It has also described how the hierarchical data-structure was implemented (Section §7.2) and how the model was integrated with a GIS (Section §7.3). This exposition of the computational model of the RHSM concludes Part II, which introduced a new surface model that seeks to address the question of scale using LOD techniques. The following and final Part III evaluates and assesses the RHSM by utilising it for geomorphometric hydrological analysis.

Part III.

Hydrological Applications

The Regular Hierarchical Surface Model introduced in Part II is assessed in this part by utilising it for a geomorphometric hydrological analysis similar to that described in Chapter 4 of Part I. Chapter 8 adapts the geomorphometric hydrological analysis for an RHSM environment. Chapter 9 evaluates the performance of the RHSM and the adapted hydrological model on a range of synthetic and LiDAR derived surfaces. Chapter 10 discusses the key findings and proposes further development of the RHSM. Chapter 11 concludes this thesis.

8. Hydrological modelling in the RHSM

This chapter adapts the geomorphometric hydrological modelling techniques that were outlined in Chapter 4 for the RHSM. The techniques are generalised to apply to triangular and hexagonal structures, where possible; and adapted to support variable resolution regular tessellations, including non-conforming edges. These techniques are evaluated in Chapter 9.

The RHSM described in Part II could represent surface information from any problem domain that features variability of parameters with scale, which arguably includes all domains that involve spatially distributed variables. However, Subsection 4.4.4 identifies urban hydrology as a field where multi-scale modelling is potentially advantageous due to the characteristics of the landscape of urban areas.

Section 8.1 examines the role of flow direction in generating RHSM sparse realisations for hydrological models. Section 8.2 outlines geomorphometric hydrological analysis on a variable density surface. The following sections: Section 8.3, Section 8.4, and Section Section 8.5 detail flow direction, variable density flow direction, and flow accumulation on the RHSM. Section 8.6 concludes the chapter with the hydrological class specification.

8.1. Resolution indicators from geomorphometry

The hydrological variables and parameters of geomorphometry outlined in Section 4.3 are an effective form of analysis of elevation data. This section summarises the role of elevation data in hydrological models (Subsection 8.1.1) and proposes that the geomorphometric parameter of flow direction (Subsection 8.1.2) is the critical parameter for determining the spatial resolution for hydrological models that involve the discretisation of the plane into subsets of space that have consistent properties.

8.1.1. Data driven resolution

As discussed in Chapter 4, DEMs provide the source data used to estimate many variables used in hydrological modelling, including slope, curvature, flow direction, flow accumula-

tion, down slope area, topographic wetness index, and surface water depth. Surface water is affected by gravity. Therefore hydrological variables are in general field specific (see Section 4.1). Elevation data can also be used to identify hydrologically important objects in the landscape such as sinks, rivers, catchments, hills, and buildings. In addition, combining elevation data with knowledge of geological and anthropogenic processes can characterise an area in the absence of other information.

Traditionally, the computational scale of geomorphometry was driven by the resolution of the DEM, which for more complex cell based methods of analysis, such as a surface water depth calculation, may be unnecessarily and impractically dense. Lumped models ameliorate this by summarising properties over a larger area to provide cohesive areas for computational modelling. Due to its pivotal role in H&H models, elevation is the logical basis to inform the resolution of the computational surface for hydrological modelling. However, it need not be elevation itself that controls resolution; it could be a variable or a combination of variables derived from elevation using the techniques of geomorphometry that informs the subdivision of space into lumped entities.

It is proposed that DEMs can provide not just the elevation data, which is a parameter in hydrological models, but also the information to determine the size of the computational units within the discretisation of space. The discretisation should be fine enough to avoid significant sub-pixel heterogeneity but coarse enough to minimise computational load and be commensurate with other measurements contributing to the model.

8.1.2. Why flow direction?

To reduce computations, we need to employ resolution indicator variables that can be calculated easily. The options from hydrological geomorphometry are elevation, flow direction, and flow accumulation. This subsection uses a 1-D surface to illustrate the importance of direction to provide a restraint to upscaling. This is extended to 2-D surfaces in Section 8.4.

Computational scale affects the determination of direction and slope in discrete space. Consider, for example, the slope of a 1-D surface in the form of a cross section or long section as in Figure 8.1. On differentiable surfaces, slope is the gradient of the tangent to the curve at a given point. However, topographical surfaces do not generally contain well defined tangents (Mandelbrot, 1977). In fact, fractal surfaces do not tend to a fixed value with finer resolution. On non-differentiable surfaces, the slope of the surface can be estimated by finding the rise over run from sample points. Unsurprisingly, the local slope estimated in this way varies depending on the spacing of the sample points. The

Nyquist-Shannon sampling theory from communication theory states that the minimum sampling rate required to accurately reconstruct a signal from discrete samples is twice the highest frequency in the signal (Jerri, 1977).

Four possible sample intervals are shown of a synthetic surface in Figure 8.1. The structure of the sampling forms an aligned 1-D hierarchy. The sample sites are covered by sample sites at finer resolutions and can be converted into an aligned 1-D tessellation of slope by assigning single slope values to the line segments between data points. The slope value at a coarser resolution is equivalent to the mean of the signed slope of its children. In this context a "signed" slope is relative to a direction. In the case of Figure 8.1, this is the direction of the x-axis.

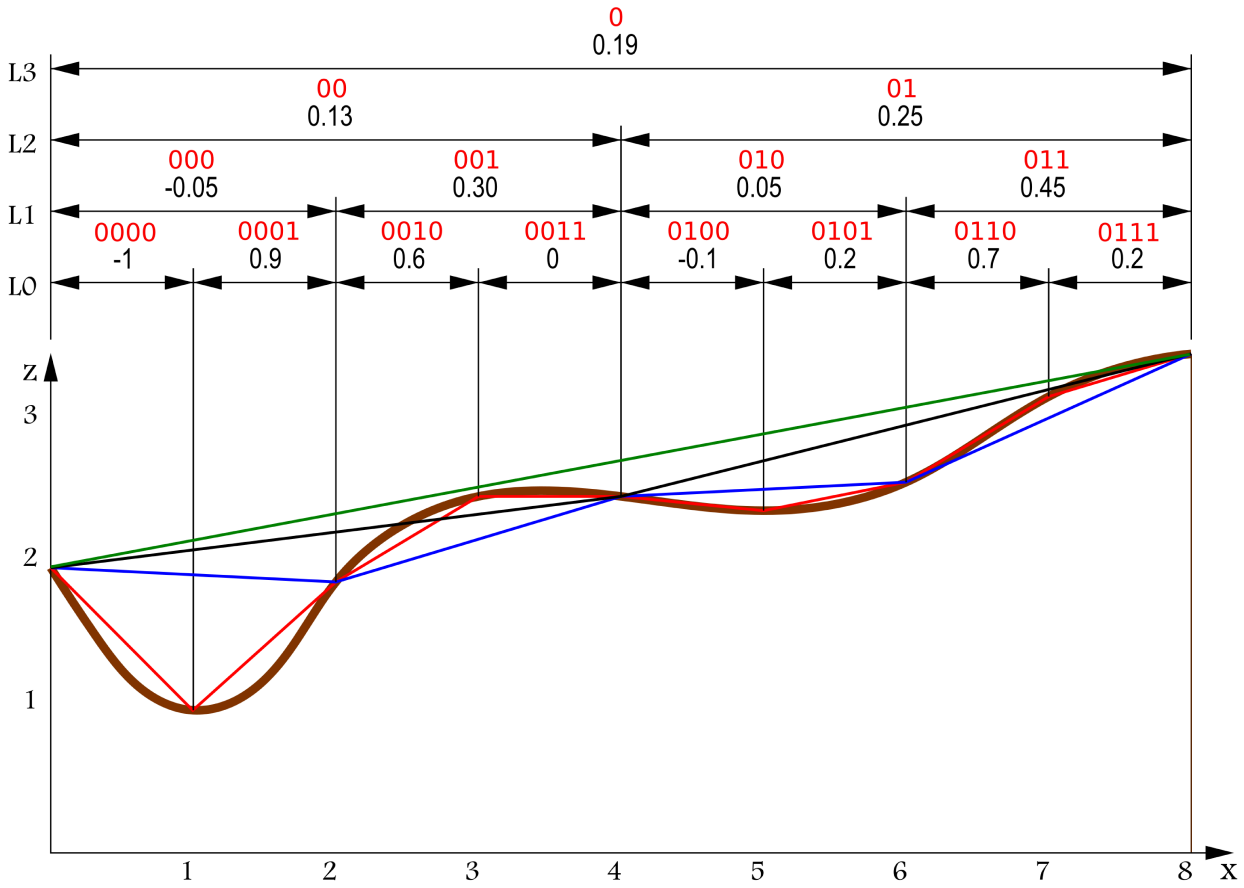


Figure 8.1.: Slope at four resolutions. Slope is calculated from the continuously varying thick brown line at four sampling resolutions. From finest to coarsest the slopes are coloured red, blue, black, and green. The steepest sections of the finest resolution (red line) coincide with the flattest part of the second finest resolution (blue line). The line segments are hierarchically indexed with a binary tree index shown with red text.

The coarse scale representations in Figure 8.1 are not always accurate representations of the underlying fine resolution slopes. For instance 0000 and 0001 are the steepest cells.

However, due to their opposing sign, their parent in L1 (000) is the equal flattest[1]. If a variable resolution surface was developed from these samples, it would need to represent 000 at the finest resolution (0000 and 0001) in order to accurately represent the slope. A decision rule could be introduced to define acceptable resolution by introducing a threshold for deviance of the parent's value from the child value; the parent is only used where the deviance is within the threshold.

For arguments sake, let us propose a deviance value of 0.3, 010 would pass and therefore be represented as 010 rather than 0100 and 0101. However, this would miss an important hydrological feature because surface water would be trapped in the shallow channel. Possibly this area is part of a channel that transmits water a significant distance. Therefore, the decision rule should consider changes in sign as important features or, in other words, flow direction should be conserved rather than (or in addition to) slope. The slope may only differ slightly in some areas, whereas, the direction is opposed. For this reason conservation of flow direction is proposed as the decision rule for sparse realisations for geomorphometric modelling as opposed to elevation, slope, or flow accumulation.

8.2. RHSM flow modelling

The hydrological applications described in this chapter are an extension of the geomorphometric hydrological analysis described in Chapter 4. Except that multiple resolutions, variable density, and three different geometries of tessellation are used. Geomorphometric flow models represent static, kinematic surface flow (see Section 2.2). This arguably removes the true complexity of hydrological flow modelling and generates results that are not easily confirmed by observation of hydrological catchments.

The proposed RHSM flow modelling has the following steps:

1. Extract surface information from observation data at base resolution.
2. Perform hydrological conditioning: sink removal and inclusion of ancillary information, such as, buildings and drainage features.
3. Determine flow direction using derivatives of $D8$ or $D\infty$ with adaptations to incorporate triangular and hexagonal geometries (Section 8.3).
4. Perform LOD analysis on flow direction by forming pyramids and extracting a variable density realisation (Section 8.4).
5. Determine flow accumulation in the variable density realisation using algorithms adapted for the purpose (Section 8.5).

[1]Due to an unintended optical illusion, 010 appears flatter but, in fact, has the same slope.

6. Determine catchment boundaries. Catchment boundaries require non-dispersive flow routing methods to prevent ambiguity; for instance, $D8$-LTD (Orlandini and Moretti, 2009).

The novel aspects that RHSM introduces into flow modelling most strongly influence steps 3, 4, and 5, therefore those steps are the focus of this chapter and the assessment in Chapter 9.

8.3. Flow direction

This section describes the adaption of $D8$ (Subsection 8.3.1) and $D\infty$ (Subsection 8.3.2) to other geometries of tessellation and also considers Rho8 optimisation (Subsection 8.3.3).

8.3.1. Dx: $D8$, $D6$, and $D3$

The $D8$ flow direction algorithm that was discussed in Subsection 4.3.1 is a neighbourhood operation utilising a 3×3 neighbourhood. $D8$ defines a single restricted direction by identifying the steepest downslope neighbour; the direction to the identified neighbour is the $D8$ flow direction. The distance to neighbours is considered in calculating the slope.

The corollaries of $D8$ in RHSM-hex and RHSM-tri are $D6$ and $D3$ respectively. $D8$, $D6$ and $D3$ are, here, collectively known as Dx. $D3$ is actually a corollary of the rectangular $D4$ because it ignores the point adjacent neighbours. The direct $D8$ corollary, $D12$, was not developed further because it lacks the self similar neighbourhoods required for the HIP referencing system. The neighbourhoods and directions for $D8$, $D6$, $D3$, and $D12$ are shown in Figure 8.2. Dx algorithms are evaluated in Section 9.2.

The flow direction can be represented in RHSM by the vector name corresponding to the direction (see Figure 8.2). The RHSM vector direction (Dx) can be converted to radians ($D_{radians}$):

$$D_{radians} = \frac{2\pi}{\mathfrak{a} - 1}\,(Dx - 1) \tag{8.1}$$

Where $\mathfrak{a}$ is the aperture of the RHSM dataset. However, for RHSM-tri the varying base vector must be allowed for.

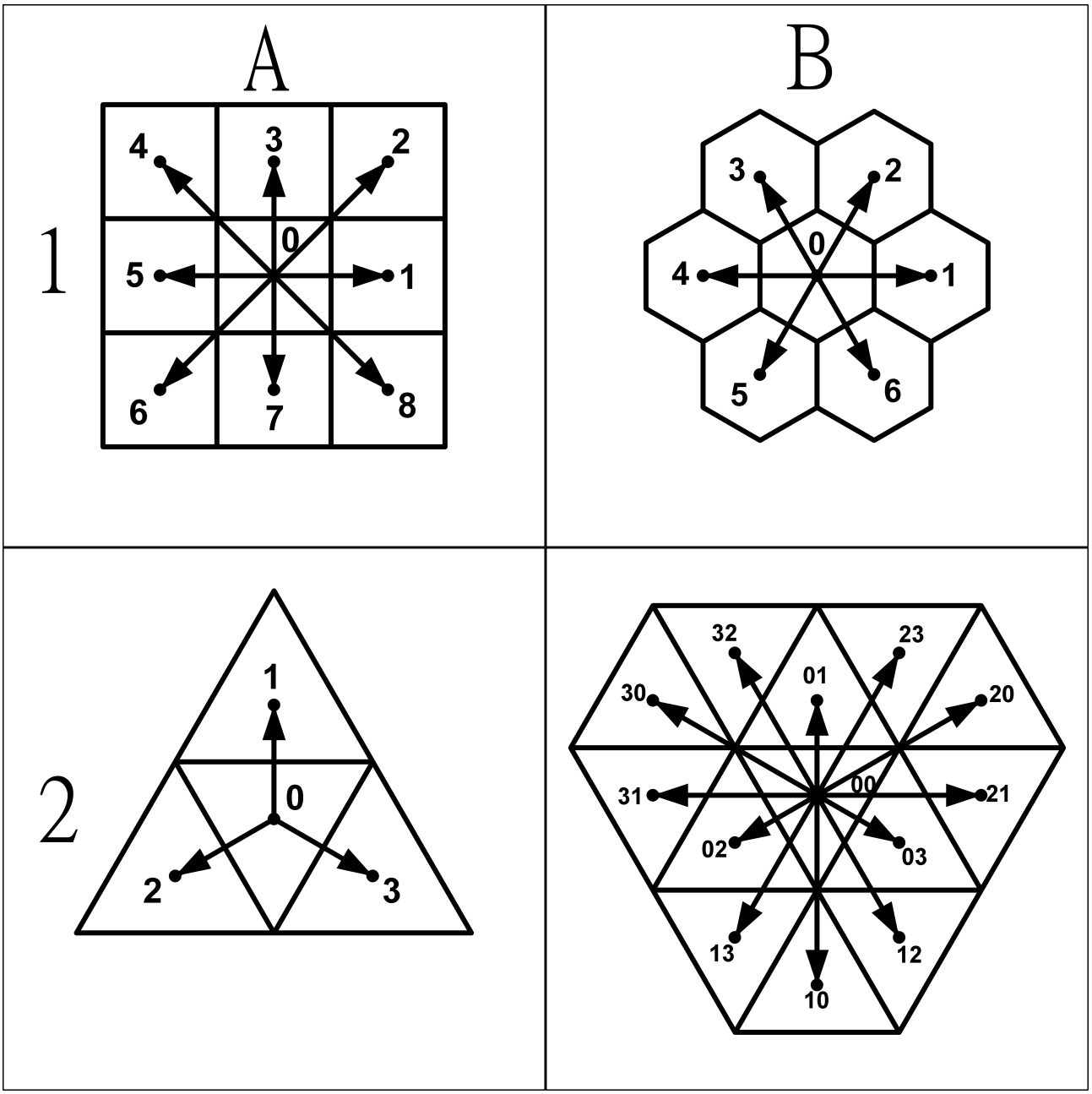


Figure 8.2.: $D8$, $D6$, $D3$, and $D12$ directions and neighbourhoods. Cells are numbered with their RHSM vector indices. A1 is $D8$, B1 is $D6$, A2 is $D3$, B2 is $D12$. $D12$ was not pursued because it lacks self similar neighbourhoods.

8.3.2. $D\infty$ in RHSM

The $D\infty$ flow direction algorithm was introduced in Subsection 4.3.1. $D\infty$ is described here in more detail as it relates to the RHSM. $D\infty$ is a neighbourhood operation utilising a 3×3 neighbourhood. $D\infty$ flow directions are determined as an angle in radians anticlockwise from the x-axis and, in the RHSM, are then converted to RHSM directions before running the flow accumulation algorithm. $D\infty$ determines single unrestricted directions utilising triangular facets formed by connecting the centre of the cell and the centres of adjacent neighbours (see Figure 8.3). The direction and slope of the line of steepest descent is calculated on each triangular facet. If the direction for a specific facet is not within the arc formed by the two edges radiating from the centre, the direction and slope for that facet is assigned to the nearest edge. $D\infty$ direction for the cell is the steepest downslope of the directions calculated for the facets.

$D\infty$ facets for rectangular, hexagonal, and triangular facets are shown in Figure 8.3. An example of a hexagonal $D\infty$ flow direction is shown in Figure 8.4. The determination of $D\infty$ in RHSM-hex is given below first as an adaptation of the method of Tarboton (1997) and then as a generalised method. The figures in this section and Section 8.5 illustrate $D\infty$ determination only in hexagonal tessellation. However, rectangular and triangular equivalents are not significantly different. The reader may also refer to Tarboton (1997) for rectangular $D\infty$.

$D\infty$ for hexagonal tessellation

$D\infty$ can be implemented in RHSM-hex by considering the facet in Figure 8.5. Following the method of Tarboton (1997) adapted for hexagonal grid structure, downward slope for a single facet is represented by the vector (S_1, S_2):

$$\begin{aligned} S_1 &= {}^{(e_0 - \frac{e_1+e_2}{2})}\big/_{\frac{\sqrt{3}}{2}d} \\ S_2 &= {}^{(e_1-e_2)}\big/_{d} \end{aligned} \tag{8.2}$$

where d is the distance between cell centres and e_x is the cell elevation as shown in Figure 8.5. The slope direction (r) and magnitude (s) are determined as:

$$r = \tan^{-1}\left(\frac{s_2}{s_1}\right) \tag{8.3}$$

$$s = \sqrt{s_1^2 + s_2^2} \tag{8.4}$$

Equation 8.3 uses the 2-parameter tangent function sometimes known as tan2. If r is not in the range $(-\pi/6, \pi/6)$, r is set as the direction of the nearest edge of the facet and s is set to the slope of that edge according to:

$$\begin{cases} r < \frac{-\pi}{6} & r = \frac{-\pi}{6},\ s = {}^{(e_0-e_1)}/_{d} \\ r > \frac{\pi}{6} & r = \frac{\pi}{6},\ s = {}^{(e_0-e_2)}/_{d} \end{cases} \tag{8.5}$$

The six facets in Figure 8.4 can be mapped to the facet in Figure 8.5 by appropriate selection of corner elevations (e_0, e_0, e_2) and rotation constant (a_c). For the facet number in Figure 8.4, Table 8.1 indicates the values of e_0, e_0, e_2, a_c to use. The multiplier required to account for the difference in distance to the edge and point adjacent neighbours in rectangular $D\infty$ is unnecessary in hexagonal $D\infty$ due to the rotational symmetry of the triangular facets in RHSM-hex. The facet with the highest slope magnitude (s) is selected as the direction angle (r'). Calculated angle r' can be converted to $D\infty$ direction measured anti clockwise from east by applying:

$$D\infty = r' + a_c{}^{\pi}/_{3} + {}^{\pi}/_{6} \tag{8.6}$$

Finally, if required, the angle can be converted to HIP ordinate by dividing by the angular difference between successive HIP values, which in the case of RHSM-hex is done using:

$$D\infty_{RHSM} = {}^{D\infty}/_{\frac{\pi}{3}} + 1 \tag{8.7}$$

If the magnitude of the slope (s) is negative, $D\infty$ is set to 0 to indicate that the flow direction is toward itself i.e. a sink.[2]

Table 8.1.: Facet elevations and factors for slope and angle calculations.

facet	1	2	3	4	5	6
e_0	0	0	0	0	0	0
e_1	1	2	3	4	5	6
e_2	2	3	4	5	6	1
a_c	0	1	2	3	4	5

[2]Tarboton (1997) uses -1 to represent sinks.

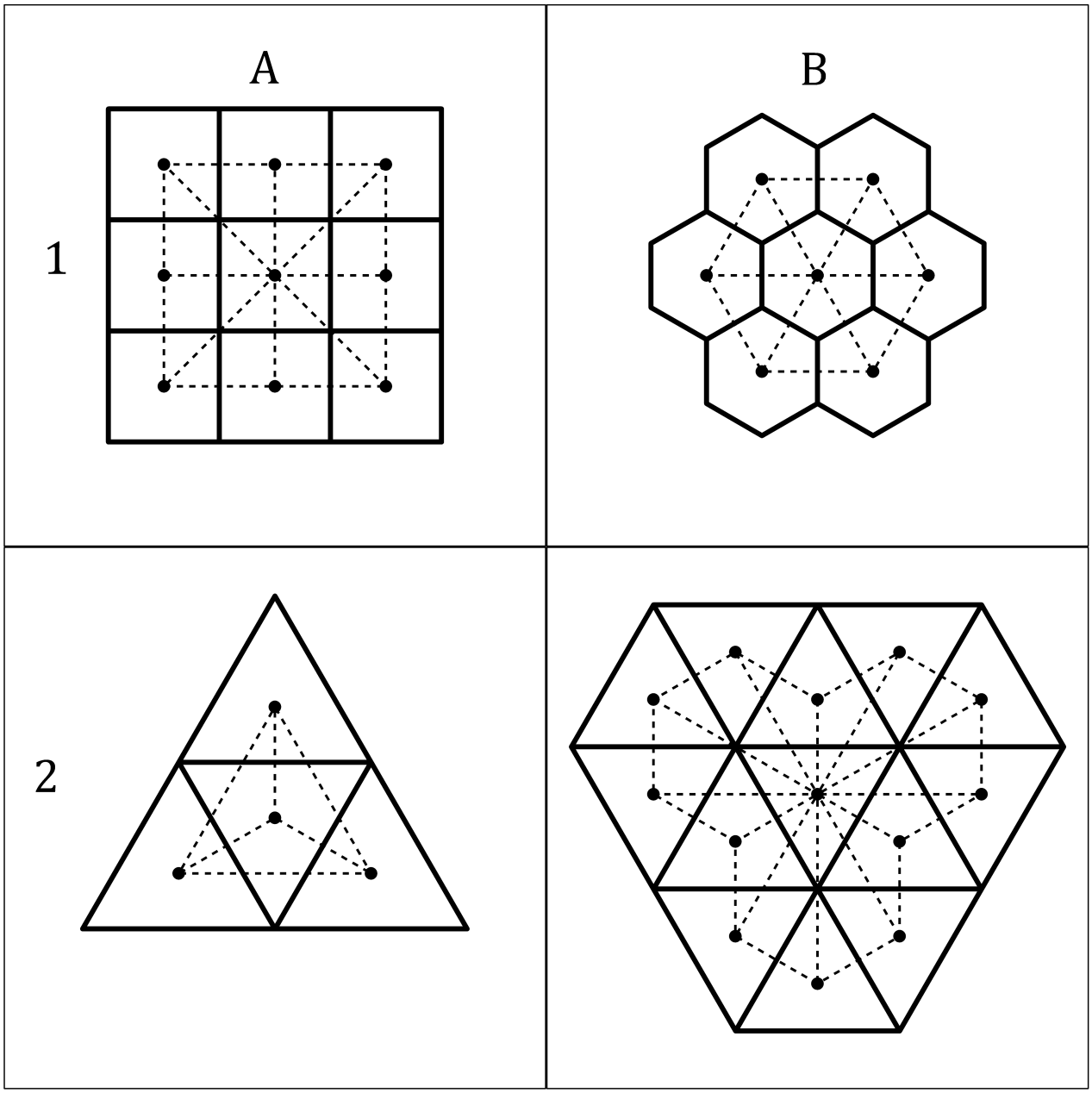


Figure 8.3.: $D\infty$ facets and neighbourhoods. The edges of the triangular facets used by $D\infty$ are shown in dashed lines. The cell edges are shown in solid lines.

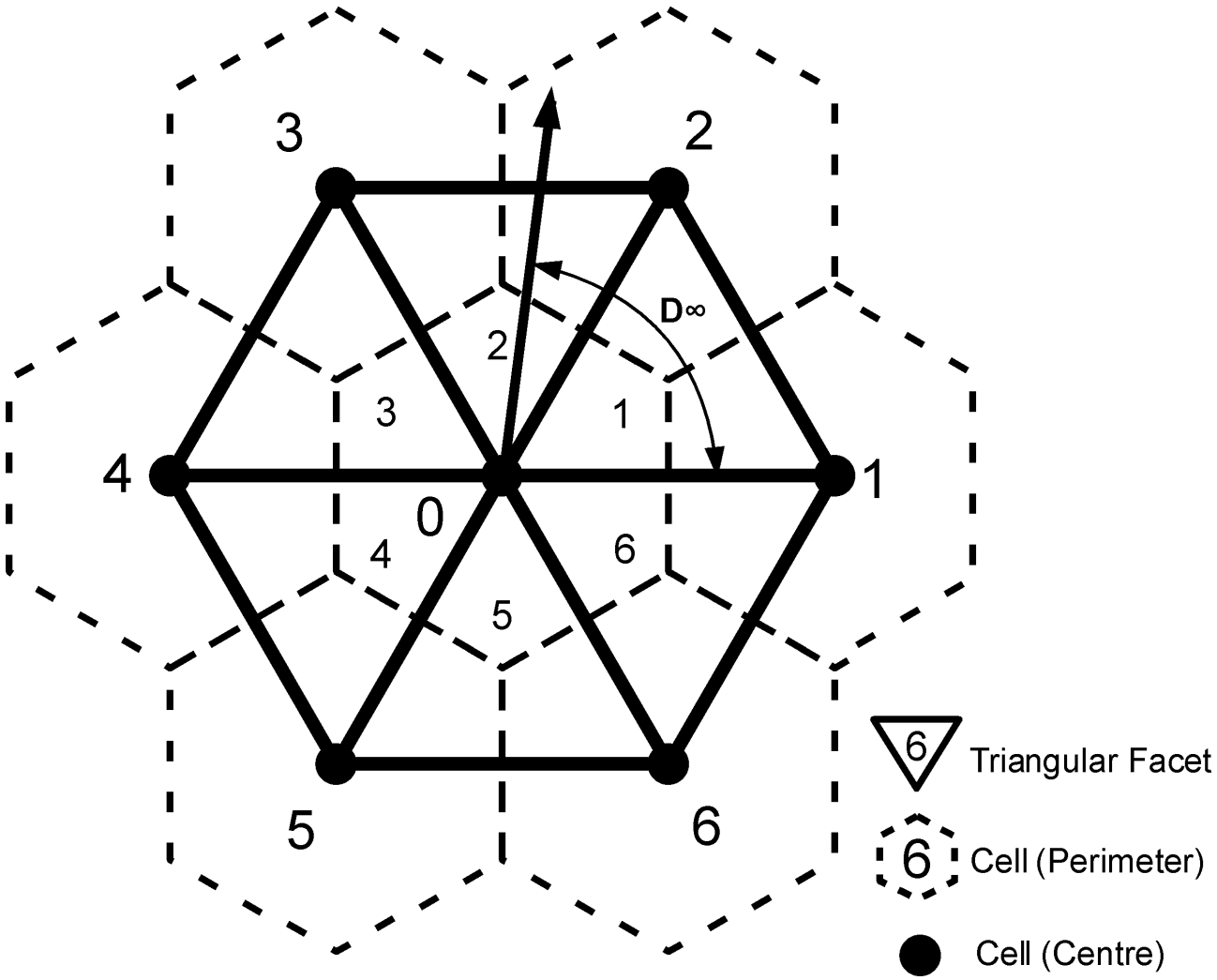


Figure 8.4.: $D\infty$ flow direction (black arrow) is defined on a hexagonal grid as the steepest downslope direction on triangular facets created by connecting the centres of the hexagonal grid cells. The heavy solid lines are the edges of the facets. Tarboton (1997) defined $D\infty$ as the anticlockwise direction in radians as shown by the thin two-headed arrow.

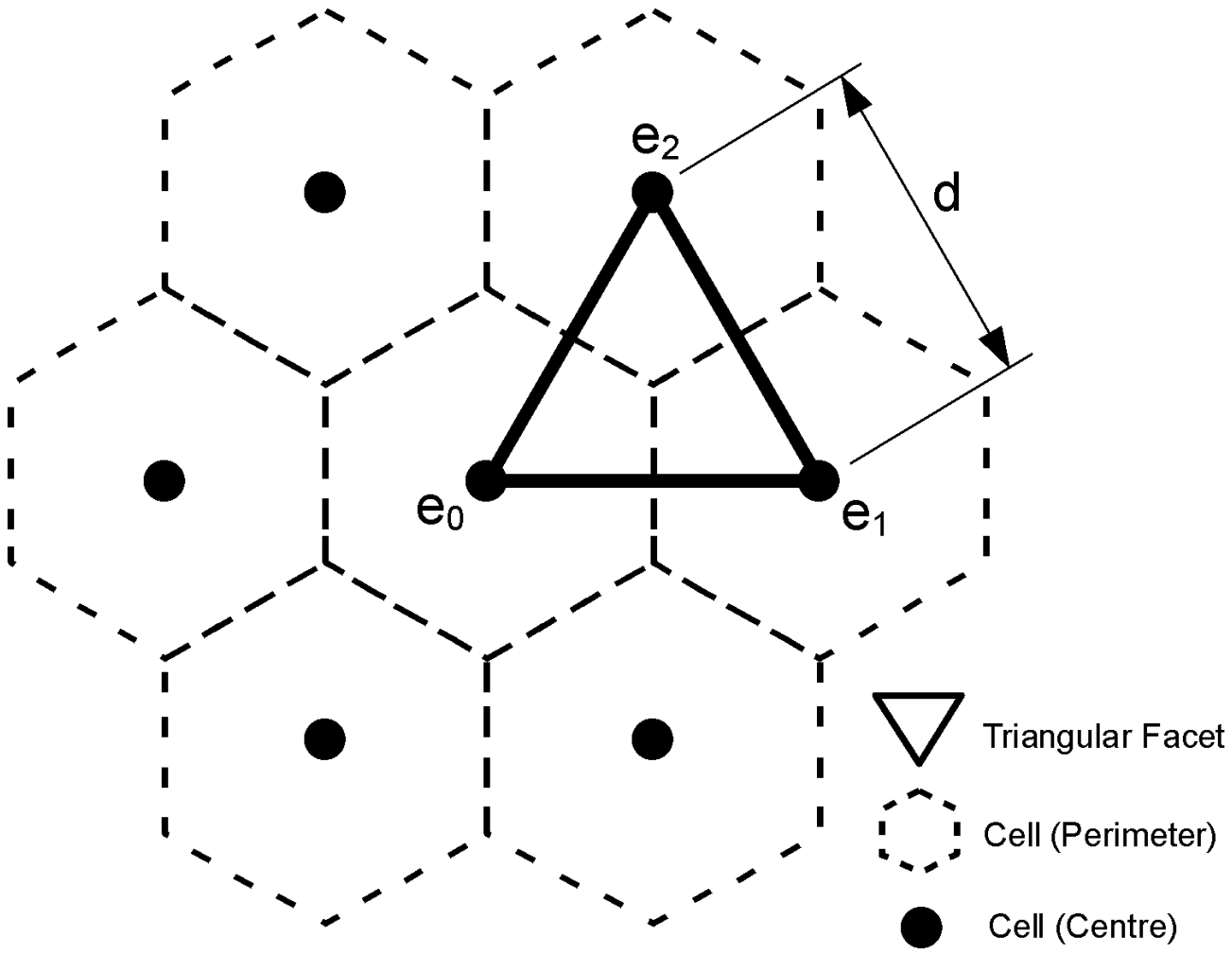


Figure 8.5.: Variables for calculation of $D\infty$ direction and slope for a single facet in RHSM-hex.

General method

The slope for a plane can be determined from any three points on that plane; therefore, $D\infty$ can be generalised for any geometry of tessellation. Using the generalised method there is no difference between the algorithms for the different geometries. Therefore they are not assigned individual names (unlike $D8$ and corollaries).

Using the notation from Figure 8.5, and given the Cartesian coordinates of point e in $\mathbb{R}^3$, (e_{0x}, e_{0y}, e_{0z}) for each triangular facet, the following equation uses the coordinates of the cell centres to create two vectors $\mathbf{u}, \mathbf{v}$ that represent the edges of the facet radiating from the centre point (e_0).

$$\begin{aligned} \mathbf{u} &= (e_{1x} - e_{0x},\, e_{1y} - e_{0y},\, e_{1z} - e_{0z}) \\ \mathbf{v} &= (e_{2x} - e_{0x},\, e_{2y} - e_{0y},\, e_{2z} - e_{0z}) \end{aligned} \tag{8.8}$$

The normal ($\mathbf{n}$) is then found by determining the cross product which is converted to a unit normal ($\hat{\mathbf{n}}$) using Equation 8.9.

$$\begin{aligned} \mathbf{n} &= \mathbf{u} \times \mathbf{v} \\ \hat{\mathbf{n}} &= \tfrac{\mathbf{n}}{\|\mathbf{n}\|} \end{aligned} \tag{8.9}$$

The direction of steepest descent on a plane is equivalent to the direction of the projection ($\mathbf{g}$) of a reference vector onto the plane. The reference vector represents the down direction. In a planar projection of geographic space, the down reference vector ($\hat{\mathbf{r}}$) is typically (0, 0, -1). The projection of a line onto a plane can be found using:

$$\mathbf{g} = \hat{\mathbf{r}} - \hat{\mathbf{n}}\,(\hat{\mathbf{r}} \cdot \hat{\mathbf{n}}) \tag{8.10}$$

The slope direction (r) and magnitude (s) are determined using:

$$r = \tan^{-1}\left(\frac{g_y}{g_x}\right) \tag{8.11}$$

$$s = \frac{g_z^2}{\sqrt{g_x^2 + g_y^2}} \tag{8.12}$$

Note that the equation for r is sensitive to the sign of the divisors. If your software produces angular results that can be both positive and negative, you need to convert the output to a positive result between 0 and 2π, for instance, by using the mod function (%) as in:

$$r' = (2\pi + r)\,\%2\pi \tag{8.13}$$

Hence forth, the use of the $'$ notation indicates that an angle has been normalise to between 0 and 2π.

If r' is outside the limits of the angular facet it is necessary to convert the direction and slope to those of the nearest edge using:

$$r'_1 = \tan^{-1}\left(\frac{u_y}{u_x}\right),\ s_1 = \frac{u_z^2}{\sqrt{u_x^2 + u_y^2}}$$

$$r'_2 = \tan^{-1}\left(\frac{v_y}{v_x}\right),\ s_2 = \frac{v_z^2}{\sqrt{v_x^2 + v_y^2}}$$

$$\triangle r'_i = \min\left(2\pi - |r' - r'_i|\,, |r' - r'_i|\right)$$

$$\begin{cases} \max\left(\triangle r'_1, \triangle r'_2\right) \geq h & \begin{cases} \triangle r'_1 \leq \triangle r'_2 & r' = r'_1, s = s_1 \\ \triangle r'_1 > \triangle r'_2 & r' = r'_2, s = s_2 \end{cases} \end{cases} \tag{8.14}$$

where $\triangle r'_1 = \triangle r'_2$, the values are arbitrarily set to the first case. However, this situation cannot be the steepest downslope direction as it only occurs when the slope direction is pointing exactly opposite from the direction of the facet.

Equation 8.14 identifies the greatest angular difference between r' and the direction of the two edges of the facet radiating from the central point (r'_1, r'_2,). If this angle is greater than the interior angle of the edges of the facet (h), the direction is "outside" the facet and should be adjusted. For RHSM-hex $h = \pi/3$, RHSM-rec $h = \pi/4$ and for RHSM-tri $h = 2\pi/3$.

The facet with the highest slope magnitude (s) is selected as the direction angle ($D\infty$) expressed as a direction in radians. Where more than one facet has the steepest slope, the smallest direction angle is assigned. It was assumed that this would occur rarely. However, Section 9.3 showed that this assumption is consequential for dispersive cone surfaces.

8.3.3. Rho8 formula optimization

$D8$ flow direction has predictable, well understood direction bias toward the $D8$ directions. $D\infty$ is one approach to reduce this bias. An alternative approach is Rho8 (Fairfield and Leymarie, 1991). Rho8 was introduced in Section 4.3.

This section suggests an alternative formulation of Rho8 that further reduces bias and provides an additional parameter to control the impact of the random variable. Rho8 attempts to reduce $D8$ direction bias by multiplying the elevation difference between the central cell and its diagonal neighbours by a random factor, ρ_8, given by Fairfield and Leymarie (1991) as Equation 4.4. $D8$ multiples the elevation difference between the central cell and diagonal neighbours by $1/\sqrt{2} \approx 0.7071$ to account for the increased distance to the diagonal neighbour. Therefore, compared to $D8$, if $\rho_8 < 1/\sqrt{2}$, it favours the cardinal (non-diagonal) neighbour, if $\rho_8 > 1/\sqrt{2}$ the diagonal neighbour is favoured, whereas, if $\rho_8 = 1/\sqrt{2}$ the $D8$ solution is invariably applied. Fairfield and Leymarie (1991) state that the expected value of $\rho_8 = 1/\sqrt{2}$. However, the expected (or median) value of $\rho 8$ ($E\rho_8$), occurs when $\sigma = 0.5$ i.e. when the following holds true.

$$E\rho_8 = \frac{1}{2 - 0.5} = \frac{2}{3} \approx 0.6666 \tag{8.15}$$

However, the expected arithmetic mean value ($\mu\rho_8$) of a large population of ρ_8 values is:

$$\mu\rho_8 = \int_0^1 \frac{1}{2 - \sigma}\, d\sigma = \log 2 \approx 0.6931. \tag{8.16}$$

Therefore $E\rho_8 < \mu\rho_8 < \frac{1}{\sqrt{2}}$, which means that $\rho 8$ generally favours the cardinal neighbours but when it does favour the diagonal neighbours it does so by a greater amount, thus raising the average. However, this effect does not fully offset the bias toward the cardinal neighbours. The value of ρ_8 does not in itself set the direction of flow for a particular cell in a raster processed by Rho8, it is merely the extent to which the diagonal or cardinal neighbour is favoured in that instance. The flow direction result depends on both the value of ρ_8 and the elevation differences. Therefore, the distribution of results from an

unbiased version of ρ_8 would be symmetrical around $\frac{1}{\sqrt{2}}$ i.e. $E\rho_8 = \mu\rho_8 = \frac{1}{\sqrt{2}}$. The obvious candidate is:

$$\rho'_8 = \frac{\sigma'}{\sqrt{2}} \tag{8.17}$$

where σ' is a symmetrically distributed random variable within the interval $1 \pm x$, $0 < x \leq 1$. Values of $x > 1$ are excluded on the grounds that this would permit uphill flows, $x = 0$ reduces to the $D8$ algorithm. The value of x needs to be sufficiently large to overcome the directional bias toward the $D8$ directions. Potential variations of ρ'_8 include uniform, normal, or student t distributions and $x = 0.5$ or $x = 1$.

Some questions arise that have not been answered here.

1. Can the cardinal directional bias of Rho8 be confirmed experimentally?
2. How well do the alternative Rho8 algorithms perform?
3. How large a value of x is required to overcome the $D8$ direction bias and is this quantity terrain dependent?
4. Does the magnitude of x influence the number of runs required to generate a consistent, bias-free mean?
5. How robust are the Rho8 algorithms to data errors?
6. How can Rho8 be adapted to other geometries?

8.4. LOD Flow direction generalisation and realisation

Having established a direction for each cell in an RHSM, whether using Dx, $D\infty$ or Rho8, the next step of the analysis is to generalise the flow direction array and realise a variable density surface. Section 8.5 addresses flow accumulation on flow direction realisations.

The LOD flow direction analysis has four parts:

1. Pyramid layers. Pyramid layers are generated using a scaling rule described below in Subsection 8.4.2, which relies on generalising directions (Subsection 8.4.1).
2. Error values. An error value is associated with each cell in each level of the pyramid. For RHSM flow direction arrays, the proposed error value is the maximum difference between the pyramid value and its finest level children.
3. Decision rule. The decision rule used to define the spatially variable resolution in the realisation is described in Subsection 8.4.3.

4. Realisation. Finally, the values associated with the coarsest resolution that meets the decision rule are stored on the appropriate branch of the sparse tree to create the variable density realisation. Further hydrological modelling is then conducted on the realisation.

8.4.1. Generalising directions

To determine variable density realisations of flow direction arrays, it is necessary to generalise directions. Generalisation of distributed values may be undertaken by determining a spatial mean for a given region. In general, the arithmetic mean of the children of an RHSM parent cell is equivalent to the spatial mean because all children are the same size. However, generalising directions involves circular arithmetic and requires care. Circular means are an example of statistics in non-Euclidian space. For instance, to determine the mean of a set of directions in radians, one needs to ensure that the equivalences of 0 and 2π, and $-\pi$ and π are respected.

Care must be taken to accurately define what is being calculated when determining circular means. There is no one right way of finding circular means but various techniques that are more or less applicable to different situations. For instance, there is a difference between turns and directions. The mean of turning 1° and turning 359° degrees is turning 180° degrees; whereas, the mean direction of the directions 1° and 359° is 0°. Also mean direction is undefined in some situations. For instance, the mean of directions 0° and 180° is 90° or 270°. The mean of 120°, 240°, and 360° is 0°, 60°, 120°, 180°, 240°, 300°, or 360°!

Three methods of determining circular means are described below. The methods below have different conceptual models and do not all generate the same results for some sets of directions.

The cosine method

The cosine method is a computationally simple way to determine mean direction that provides reasonable results. The cosine method treats angles as unit vectors, converts them to Cartesian vectors and then finds the mean of these vectors. Given the angles

$\alpha_1, ..., \alpha_n$ the angular mean can be calculated as:

$$\bar{\alpha} = \tan^{-1}\left(\frac{\sum_{j=1}^{n} \sin \alpha_j}{n}, \frac{\sum_{j=1}^{n} \cos \alpha_j}{n}\right) \tag{8.18}$$

However, this method treats angles as vectors, which for some applications is unsound and produces unintuitive results. For instance, using Equation 8.18 the mean of 0°, 0° and 90° equates to 26.565° not 30°. The cosine method is not a true arithmetic mean because the sum of the differences between the result and the sample set is not 0. The difference between the arithmetic mean and the cosine method increases with the spread of angles.

The Mitsuta method

The Mitsuta method of finding mean direction is commonly used to generalise wind direction. The Mitsuta method starts with an initial direction and then determines the amount each subsequent direction is turned from that direction. The turns are then averaged and the result added to the original direction. However the Mitsuta method does not work if the range of directions exceeds 180° (Mori, 1987). In addition, the Mitsuta method can give different results depending on the order of the directions and is known to fail if the standard deviation of the direction samples is greater than 25°.

Minimum variance method

The Minimum Variance method (MV) is a more robust technique that finds the arithmetic mean ($\bar{\alpha}$) by minimising $(\alpha_j - \bar{\alpha})^2$. That is, MV finds the most likely value by ordinary least squares. However, to appreciate the challenge of finding a least square estimate of a circular mean, consider Figure 8.6 on page 180, which depicts a set of directions $\alpha_1, ..., \alpha_n$. If the mean of the directions clustered around 0° is anticlockwise of zero, the 180° direction pulls the mean further anticlockwise and all the directions clockwise of 0° (α_{cw}) should be included in the mean as ($\alpha_{cw} + 360°$).If the mean is greater then zero, it pulls the mean clockwise and the directions anticlockwise of 0° (α_{acw}) should be included in the mean as ($\alpha_{acw} - 360°$). In this case, there is only one direction that exhibits this ambiguous behaviour; however, if the directions are evenly spread, the only way to determine the true minimum variance mean is to calculate the variance of all possible combinations. Given that each angle has two forms: α_{acw} and $\alpha_{cw} \pm 360°$, which affect the mean differently, we

can determine which version of each angle to use by determining which version minimises the variance of $\alpha_1, ..., \alpha_n$.

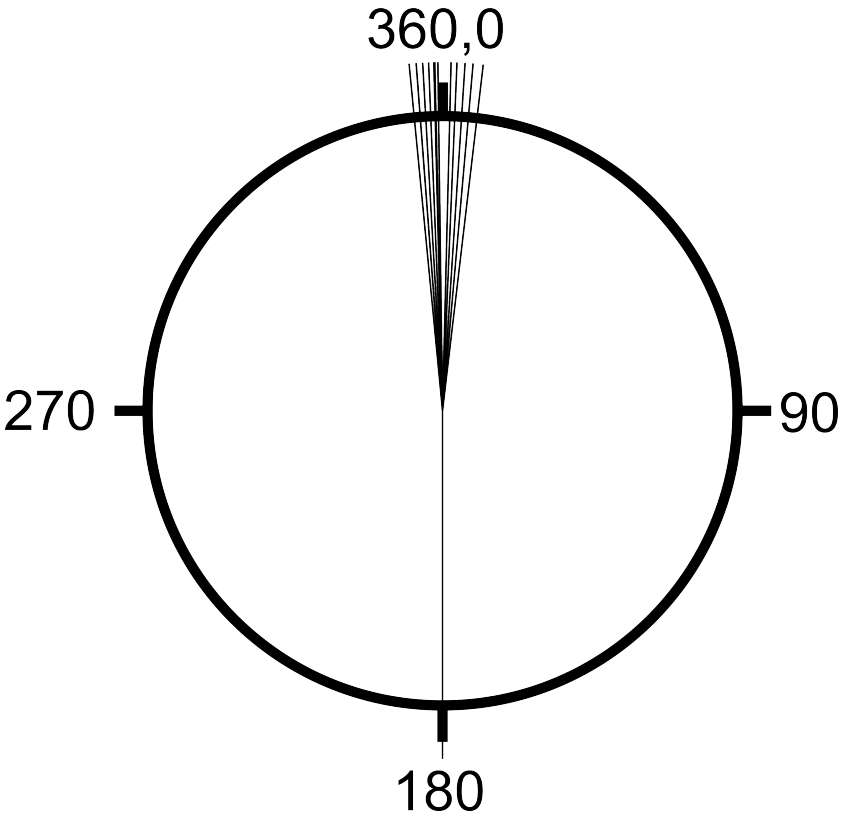


Figure 8.6.: The challenge of circular means. If the mean of the directions clustered around 0° is clockwise of 0°, the 180° direction will pull the mean further clockwise. If the mean is anticlockwise of 0°, it will pull the mean further anticlockwise.

Determining the circular mean by minimising variance is computationally intensive if determined such that the result is robust to a wide spread of angles. Fortunately, the task of generalising flow direction arrays allows us to make an important assumption that simplifies the calculation. We can assume that the spread of directions is less than 180°. This assumption is sound because it is clearly unacceptable to generalise opposing directions with a direction that resembles neither, therefore the generalised pyramid cell that contains angles with such a spread will never be used in a realisation.

If it were necessary to determine means for widely spread directions, an alternative method could be used when the spread exceeds 180°. For instance, an alternative method to deal with widely spread directions would be to divide the directions into sectors and then assign the mode sector as the generalised value.

The RHSM implementation of Restricted Minimum Variance (RMV) method generalises the circular mean to any interval: radians $[0, 2\pi]$, degrees $[0, 360]$, or HIP indices $[1, \mathfrak{a}]$, is robust for signed values and values outside the interval. If the range of the input array

is greater than half the interval, this method is unreliable, therefore, the RMV method should be employed along with a decision rule limiting the spread to below half the interval to prevent erroneous mean values being used in models.

The RMV method assumes that either all the values closest to the upper interval (α_{acw}) must be moved by subtracting the interval or none of them should be moved. This assumption is sound if the spread is less than half the interval. If the spread is less than half the interval and is contained within the interval, no values need be moved. If the range is only less than half the interval if it includes the limits of the interval, the values closest to the upper interval (α_{acw}) are moved if moving them decreases the variance. The final step is to return the final mean to the interval.

8.4.2. Pyramid layers

Pyramid layers are produced using a scaling rule. For a flow direction RHSM, pyramid values are determined by finding a circular mean using the restricted minimum variance (RMV) method described in Subsection 8.4.1.

If RHSM-hex directions are represented using RHSM values it is necessary to allow for level rotation. Each level rotates by $\tan^{-1}\sqrt{3}/2$. The difference between each direction vector is $\tan^{-1}\sqrt{3}$ therefore rotation of the base vector (rot) for a given pair of levels (L_2, L_1) can be calculated in radians as:

$$rot = (L_2 - L_1)\tan^{-1}\sqrt{3}/2 \tag{8.19}$$

or in RHSM units as:

$$rot_{RHSM} = (L_2 - L_1)\frac{\tan^{-1}\sqrt{3}/2}{\tan^{-1}\sqrt{3}} \tag{8.20}$$

where L_2 and $L_1$are represented by the level numbers.

The rotation is particularly problematic for $D6$ as opposed to $D\infty$. The $D6$ flow direction algorithm is complicated by the angle of rotation between HIP levels because the directions of the neighbours at a given LoD are not represented in the neighbourhoods of coarser and finer LoDs. Section 9.4 addresses this issue by adjusting the direction to the nearest valid direction in the coarser LoD.

8.4.3. Decision rule

A decision rule is defined and evaluated to determine the resolution to use in the variable density realisation for a given region of the tessellation. For a flow direction RHSM, the decision rule is that the error value is less than a user defined threshold number. For RHSM flow direction arrays, the proposed error value is the maximum difference between the pyramid value and its finest level children. Applying such a rule is similar to a quad-tree division of space.

The threshold number represents the greatest permissible spread of directions for a cell and provides a restraint on upscaling. The threshold value must be less than π radians otherwise the assumptions behind the RMV method are unsound. The threshold value could itself be spatially variable. For instance, a second condition could be applied that the direction threshold decreases with increasing flow accumulation.

It is possible to apply the simplification algorithm at various stages of the hydrological analysis. For instance, elevation, flow direction, or flow accumulation arrays could be simplified. This study aims to preserve hydrological significance by applying the decision rule to the flow direction array in order to minimise cell numbers without losing accuracy in representing hydrological behaviour.

There is an underlying assumption in the application of the decision rule that the direction for a region approaches a defined limit as the size of that region decreases. If that is not the case, and arbitrarily precise information is available, the decision rule would never be met. In such instances the finest resolution data are used by default.

8.5. Flow accumulation

Flow accumulation algorithms on tessellated surfaces were discussed in Subsection 4.3.2; these algorithms are well defined and are broadly applicable to alternate geometries of tessellation. Subsection 8.5.1 investigates what flow accumulation algorithms actually measure, in order to allow comparison with analytically defined Specific Catchment Area. Subsection 8.5.2 describes three methods of apportioning flows using $D\infty$. A significant difference between a flow accumulation array calculated on a single resolution and one calculated on a variable resolution realisation of a LOD model is the interaction between different LoDs. Some options for dealing with these inter-level interactions are presented in Subsection 8.5.3.

Flow direction algorithms rely on two assumptions:

1. The kinematic assumption that the flow direction must always be downhill.

2. The conservation of area. The flow accumulation of an outlet should not be more than nor less than the total area of the catchment or at least the nearest cell size unit. In reality there may be small discrepancies between the analogue shaped catchment and the digital version but the planimetric area of a catchment in its digital representation should equal the flow accumulation of the cell at its outlet.

The RHSM-tri is not compatible with $D\infty$ because the triangular facets have an internal angle of greater than 90°, which may cause a portion of flow to be directed uphill thus creating a looped flow path and infinite accumulations. Apportioning flow to an uphill cell violates the core assumption of kinematic hydrological modelling, that flows are governed by gravity. For this reason, RHSM-tri flow accumulation is not discussed after this subsection. However, one possible solution to the problem of infinite flows in RHSM-tri flow accumulation is solving the $D\infty$ direction using the $D3$ neighbourhood as described in Section 8.3 but applying the accumulation using $D12$ neighbourhoods (see Figure 8.3).

8.5.1. What does flow accumulation measure?

It is necessary to know what flow accumulation algorithms determine in order to assess their accuracy by comparison with analytically defined results. One target for comparison is Specific Catchment Area (a) as defined by Gallant and Hutchinson (2011) and discussed in Subsection 4.3.2. It is possible that some flow accumulation algorithms could derive an estimate for a at the centre of the cell directly. However, typically flow accumulation algorithms do not determine a, instead they determine the upslope area associated with an areal cell.

The $D8$ flow accumulation algorithm is conceptually very clear about what it measures. $D8$ determines the number of upstream cells linked to a given cell by the $D8$ network. $D\infty$ is more opaque; generally speaking, $D\infty$ measures the quantity of cells or part cells that can contribute flow to a given cell using the $D\infty$ definition of flow paths. But how does flow accumulation relate to specific catchment area? One approach is to assume $D\infty$ flow accumulation ($A_{D\infty}$) is equivalent to upstream area (A) and the cell size (w_c) is equivalent to the contour length (w), thereby, producing an estimate for a:

$$a \approx \frac{A}{w} \approx \frac{A_{D\infty}}{w_c} \tag{8.21}$$

However, there are several debatable assumptions in this interpretation. Flow accumulation and upstream area are not exactly the same. Upstream area is defined as the area

upstream of a length of contour. In contrast, flow accumulation is the area contributing flow to the area represented by a cell in a tessellation. Reducing a cell to a contour ignores the areal nature of the cell. For instance, it is possible that the dimension of the cell along the flow line is also important. a will also vary in 2-D across the cell, whereas, Equation 8.21 produces a point value. Therefore, another interpretation is that the flow accumulation is the integration of a along the flow path through the cell.

The assumption that contour section is equal to cell size assumes that the contour is linear or nearly so. Typically, the contour curvature is unknown so is merely assumed to be linear. In highly convergent or divergent terrains, the linear contour assumption is unsound. However, the employment of LOD analysis based on flow direction will preserve the validity of this assumption because a cell which does not include subpixel variation of direction will also not include significant contour curvature.

It may be tempting to apply a scaling factor to the flow accumulation to bring the result in line with a. However, due to the assumption of conservation of area, it is important that such a scaling factor is recorded as an adjustment to the estimation of the contour length, as adjusting the area directly would undermine the conservation of area. A further complication is that, in addition to whether the contour curvature is considered, there are at least four possible interpretations for how the cell size should be converted into the contour length:

1. The cell size without adjustment; effectively the cell is approximated as a circle and the flow width is held constant. Gallant and Hutchinson (2011) assumed that flow width is constant with respect to aspect for $D\infty$. This assumption is more sound in hexagonal sampling than rectangular sampling.
2. The cell boundary projected onto the contour that passes through the centre.
3. The intersection of the cell and the contour that passes through the centre.
4. The area of the cell divided by the length of the cell perimeter projected on the flow line, which, for simplicity, is assumed to be straight.

Of these four interpretations 1. assumes no relation between cell projection and flow accumulation, 2. and 3. assume positive correlation and 4. assumes negative correlation. These questions of what flow accumulation measures and what adjustments are required to the cell width are addressed for cone surfaces in Section 9.3.

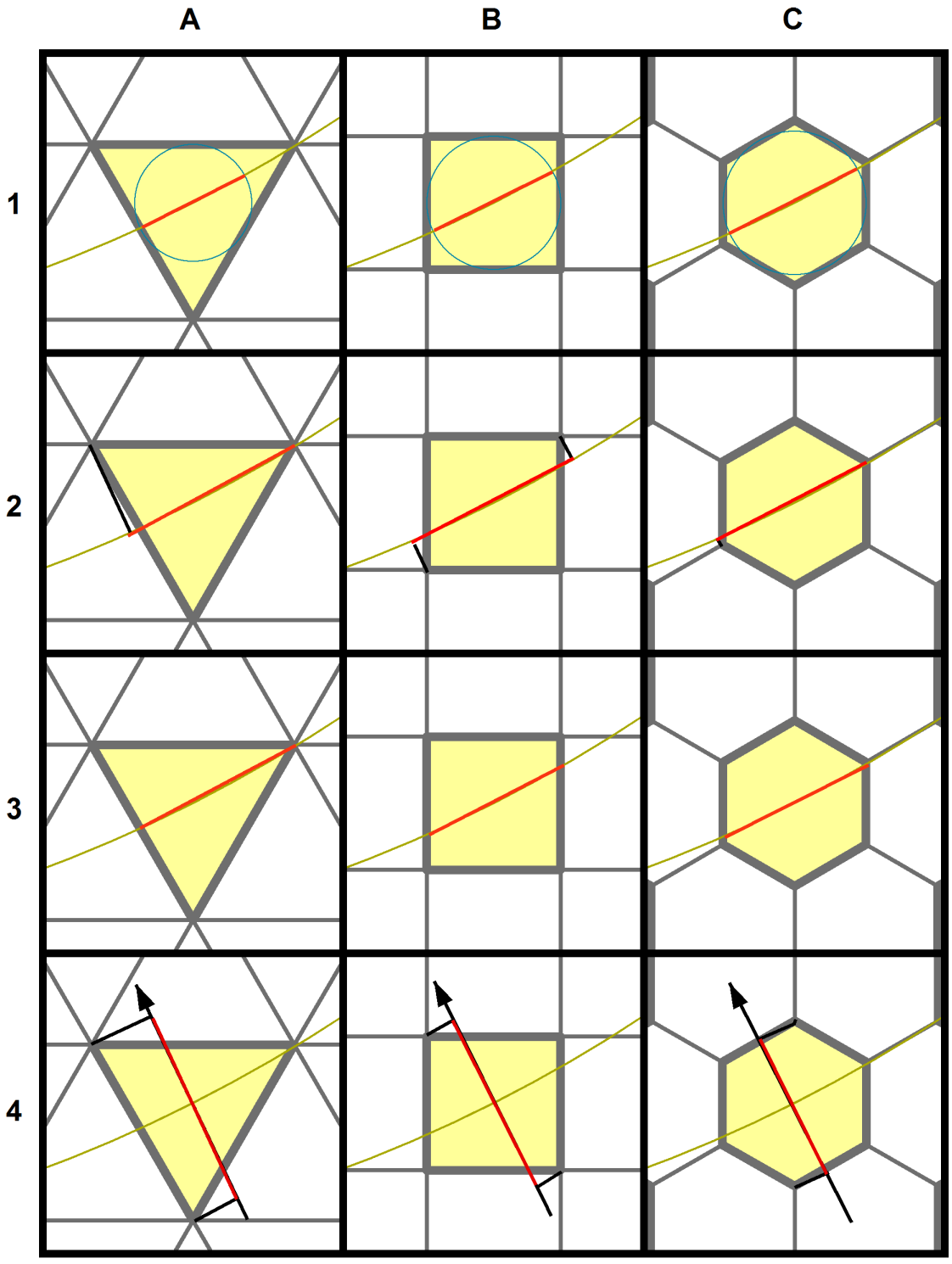


Figure 8.7.: Interpretations of cell width. For a given cell geometry (yellow tiles), several estimations of contour length are shown as a red line. The actual contour passing through the centre of the cell is shown as a green line. The four rows correspond to the four options in Subsection 8.5.1. Option four shows the flow length in red, which is the divisor of the area in this formulation and thus negatively correlated to contour estimate.

8.5.2. Angular, linear, and areal ratios

Given a single unrestricted direction there are a number of ways that flow can be apportioned between downslope neighbours, three of which are highlighted below. $D\infty$ methods are restricted to two neighbours; $D\infty$ apportions flow to only the cell centres that are closest to a line emanating from the centre of the cell in the direction of flow. If such a line intersects a neighbour's cell centre, that cell receives the entire flow.

Following mathematical convention, $D\infty$ is recorded as a radian direction anti-clockwise from the x-axis. In RHSM systems, flow direction can conveniently be represented by a decimal ($D\infty_{RHSM}$) in which the integer component (i) matches the HIP vector of the clockwise neighbour toward which it is flowing ($i = 2$ in the example shown in Figure 8.8) and the fractional part is the proportion of flow credited to the anti-clockwise neighbour (α_1). The proportion of flow allocated to the clockwise neighbour (α_2, $i = 3$ in Figure 8.8) can be found using:

$$\alpha_2 = 1 - \alpha_1 \tag{8.22}$$

$D\infty_{RHSM}$ is convenient because the apportionment algorithm can thus be abstracted away from both the direction algorithm and the accumulation algorithm. A given cell will receive all the flow from a neighbouring cell[3] in HIP direction i, if the $D\infty_{RHSM}$ of the neighbouring cell is the HIP negation ($\ominus$) of i, The proportion of flow (p_i) received from a neighbour in direction i can be accumulated as :

$$\begin{aligned} r_i &= \left| i - D\infty_{RHSM(i)} \right| \\ p_i &= 1 - \min\left(r_i'', 1\right) \end{aligned} \tag{8.23}$$

where $D\infty_{RHSM(i)}$ is $D\infty_{RHSM}$ of the cell in direction i. If $D\infty_{RHSM(i)}$ directs flow away from the cell being accumulated, r_i will be greater than 1 and that cell will contribute no flow to this cell. In order to close the circular subtraction, r_i is normalised to the range $[-1, \mathfrak{a} - 2]$ using:

$$\begin{cases} r_i > (\mathfrak{a} - 2) & r_i'' = \mathfrak{a} - 1 - r_i \\ r_i \leq (\mathfrak{a} - 2) & r_i'' = r_i \end{cases} \tag{8.24}$$

[3]To be clear, the flow direction governs flow out of the cell. Therefore, the flow received by a cell is governed by the flow directions of its neighbours. The flow direction of the cell itself plays no part.

″ is used because the range differs from that defined for r' in Subsection 8.3.2. The different range is required to make the algorithm robust to values near the $1 = \mathfrak{a}$ equality. The accumulation of flow received from direction i (Q_i) is determined as:

$$Q_i = Q_{0_i} p_i \tag{8.25}$$

where Q_{0_i} is the total accumulation for the cell in direction i.

$D\infty$ can be converted into $D\infty_{RHSM}$ using the following procedure. Here i is the integer component of $D\infty_{RHSM}$.

$$\begin{aligned} \gamma &= \tfrac{2\pi}{\mathfrak{a}-1} \\ i &= \lfloor \tfrac{D\infty}{\gamma} + 1 \\ \beta_1 &= D\infty - (i-1)\gamma \\ \beta_2 &= \gamma - \beta_1 \\ \alpha_1 &= f(\beta_1, \beta_2) \\ D\infty_{RHSM} &= i + \alpha_1 \end{aligned} \tag{8.26}$$

$\mathfrak{a}$ is the aperture of the tessellation. β_1 and β_2 are the angles formed between the $D\infty$ direction and the clockwise and anticlockwise facet edges respectively. For, example, in Figure 8.8 $(\beta_1, \beta_2) = (\alpha_1, \alpha_2)$.

$f(\beta_1, \beta_2)$ can be varied to affect different apportionment algorithms. Three variations of $f(\beta_1, \beta_2)$ are detailed below: Angular, Linear, and Areal.

Angular

The classic $D\infty$ approach from Tarboton (1997) is to apportion the flow between the two neighbouring recipients in proportion to the ratio of the angles subtended between the flow line and the edges of the facet in accordance with Figure 8.8. The proportion of flow to cell 2 is $\alpha_2/(\alpha_1+\alpha_2)$. The proportion of flow to cell 3 is $\alpha_1/(\alpha_1+\alpha_2)$. The determination of α_1 is trivial:

$$\alpha_1 = f(\beta_1, \beta_2) = \beta_1 \tag{8.27}$$

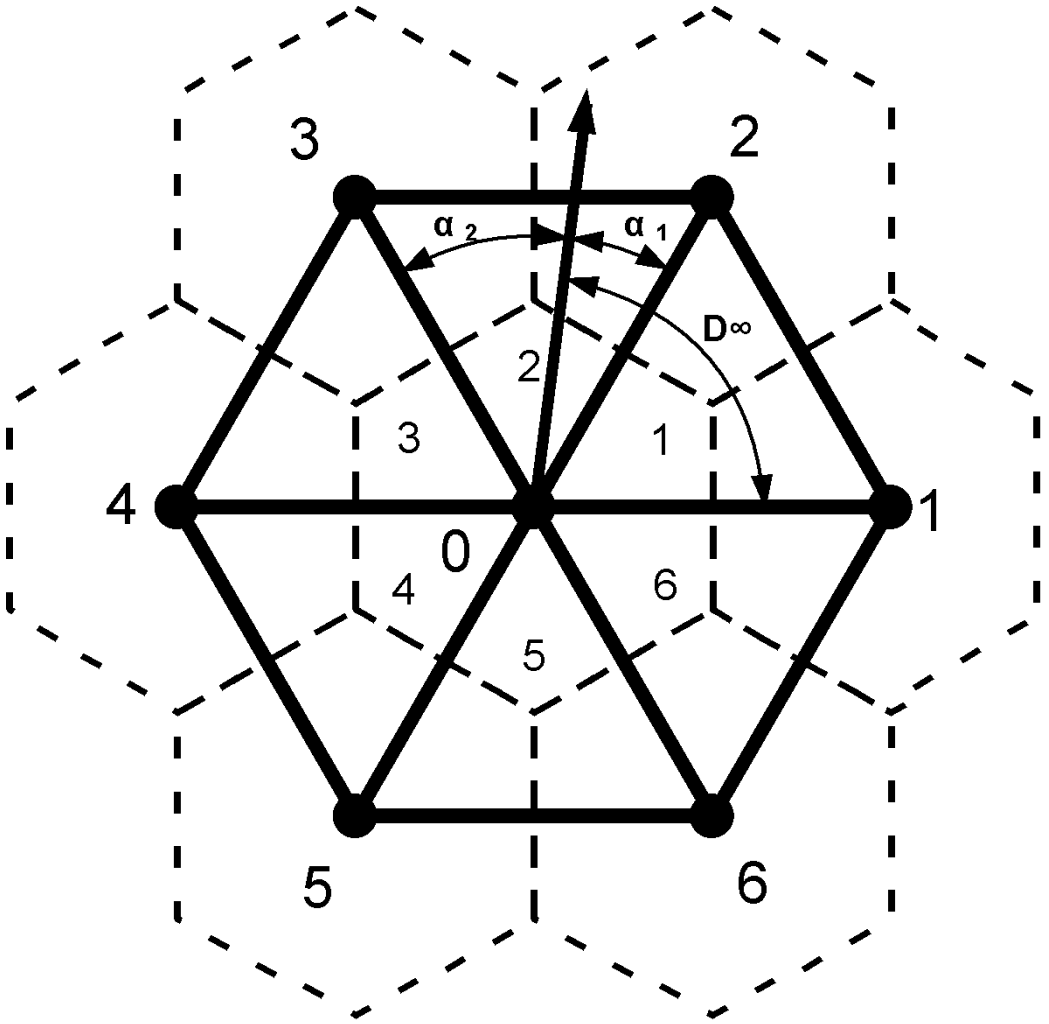


Figure 8.8.: Angular $D\infty$ apportionment for RHSM-hex.

Linear

$D\infty$ accumulations can be determined by using linear ratios instead of angular ratios. This is equivalent to the linear displacement of Orlandini and Moretti (2009). The linear method apportions flow between the two receiving neighbours according to the ratio of the linear distance between the flow direction and the edges of the facet (see Figure 8.9).

$D\infty_{RHSM-hex}$ with linear weighting is calculated using:

$$\alpha_1 = f(\beta_1, \beta_2) = \frac{\sin(\beta_1)}{\sin(\beta_1) + \sin(\beta_2)} \tag{8.28}$$

Assuming $D\infty$ is in radians anti-clockwise from the x-axis.

$D\infty_{RHSM-rec}$ with linear weighting needs to account for the varying distances to adjacent

cells. This can be done using:

$$\begin{cases} i = 0, 2, 4, 6 & \alpha_1 = \frac{\sin(\beta_1)}{\sin(\beta_1)+\sqrt{2}\sin(\beta_2)} \\ i = 1, 3, 5, 7 & \alpha_1 = \frac{\sqrt{2}\sin(\beta_1)}{\sqrt{2}\sin(\beta_1)+\sin(\beta_2)} \end{cases} \tag{8.29}$$

Note that i is the index of the triangular facet that the direction falls in and the integer component of $D\infty_{RHSM}$.

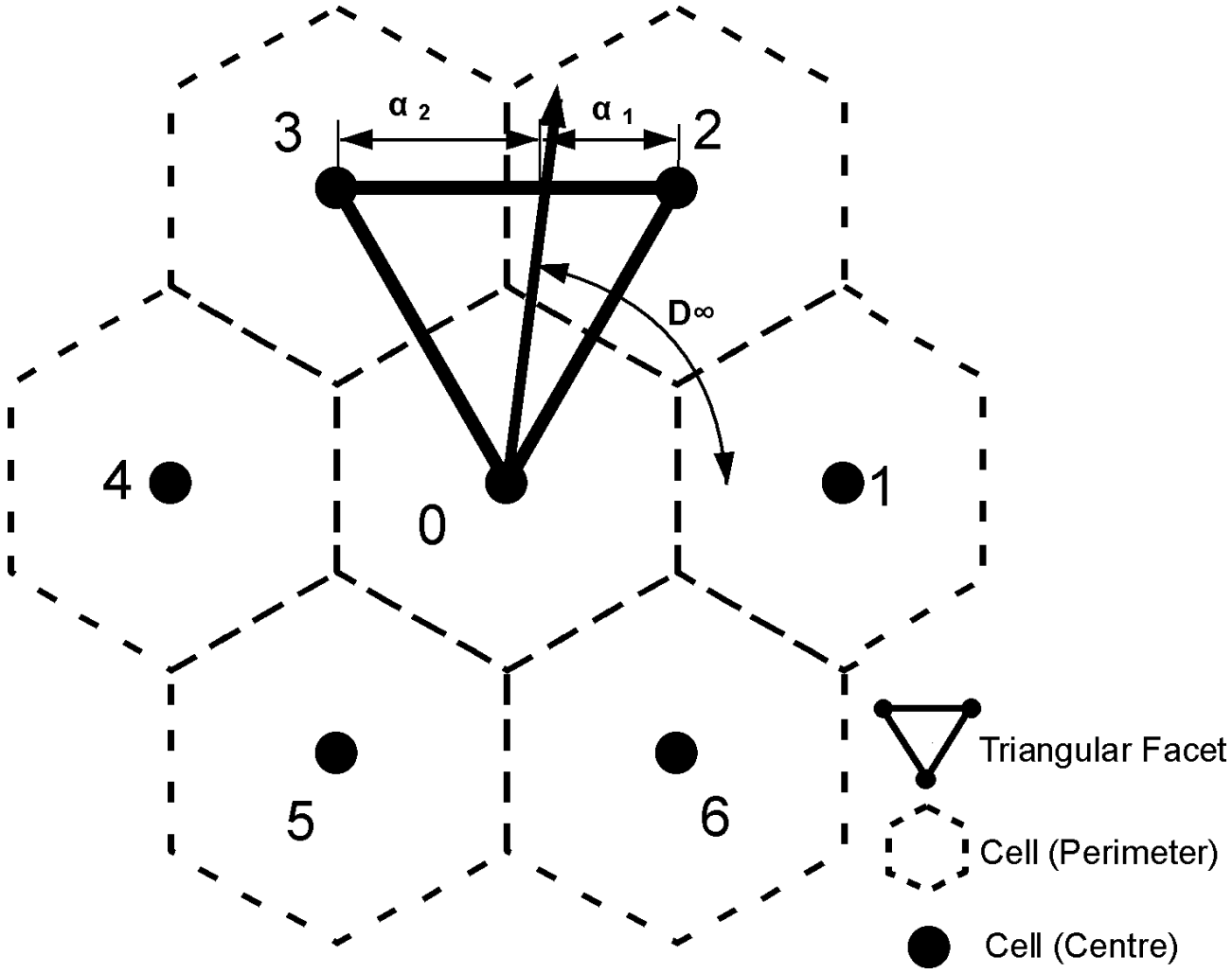


Figure 8.9.: Linear D∞ apportionment for RHSM-hex.

Areal

A third option is to also consider the length of the flow line by forming a ratio of the areas of the triangles in Figure 8.10 on page 191:

$$\alpha_1 = f(\beta_1, \beta_2) = \frac{\sin(\beta_1)\cos(\beta_1)}{\sin(\beta_1)\cos(\beta_1) + \sin(\beta_2)\cos(\beta_2)} \tag{8.30}$$

However, for RHSM-rec the diagonal neighbours need to be weighted, therefore, $D\infty_{RHSM-rec}$ with areal weighting is calculated using:

$$\begin{cases} i = 0, 2, 4, 6 & \alpha_1 = \frac{\sin(\beta_1)\cos(\beta_1)}{\sin(\beta_1)\cos(\beta_1)+\sqrt{2}\sin(\beta_2)\cos(\beta_2)} \\ i = 1, 3, 5, 7 & \alpha_1 = \frac{\sqrt{2}\sin(\beta_1)\cos(\beta_1)}{\sqrt{2}\sin(\beta_1)\cos(\beta_1)+\sin(\beta_2)\cos(\beta_2)} \end{cases} \tag{8.31}$$

8.5.3. Variable resolution accumulation

Variable resolution realisations of RHSM arrays contain non-conforming edges, for which the $D\infty$ flow accumulation algorithm must be adapted to accommodate. There are two basic scenarios; flow could be directed from coarse to fine, or fine to coarse. There are several possibilities to deal with flow from coarse to fine. Flow from coarse to fine could be evenly applied to all child cells, it could be directed only to those child cells that are neighbours of the coarse resolution cell, or it could be applied to only the child cell with the lowest elevation. The method employed will have different consequences in different neighbourhood geometries. The flow accumulation method should be developed to match the physical process being modelled as closely as possible while also producing accurate results on controlled surfaces.

Figure 8.11 shows the options considered for a $D6$ direction flowing from coarse to fine resolution. "All" evenly divides the total around all children of the recipient cell. "Centre" apportions all flow to the centre child of the recipient cell. "Leading" evenly distributes flow to the recipient's children that share a linear segment of their perimeter with the contributor. "Leading low" apportions flow entirely to the lowest elevation child that shares a linear portion of its perimeter with the contributing parent. "Low" apportions flow only to the lowest elevation child of the recipient. Leading and leading low were chosen for further development because flow does not jump cells. Figure 8.11 shows how

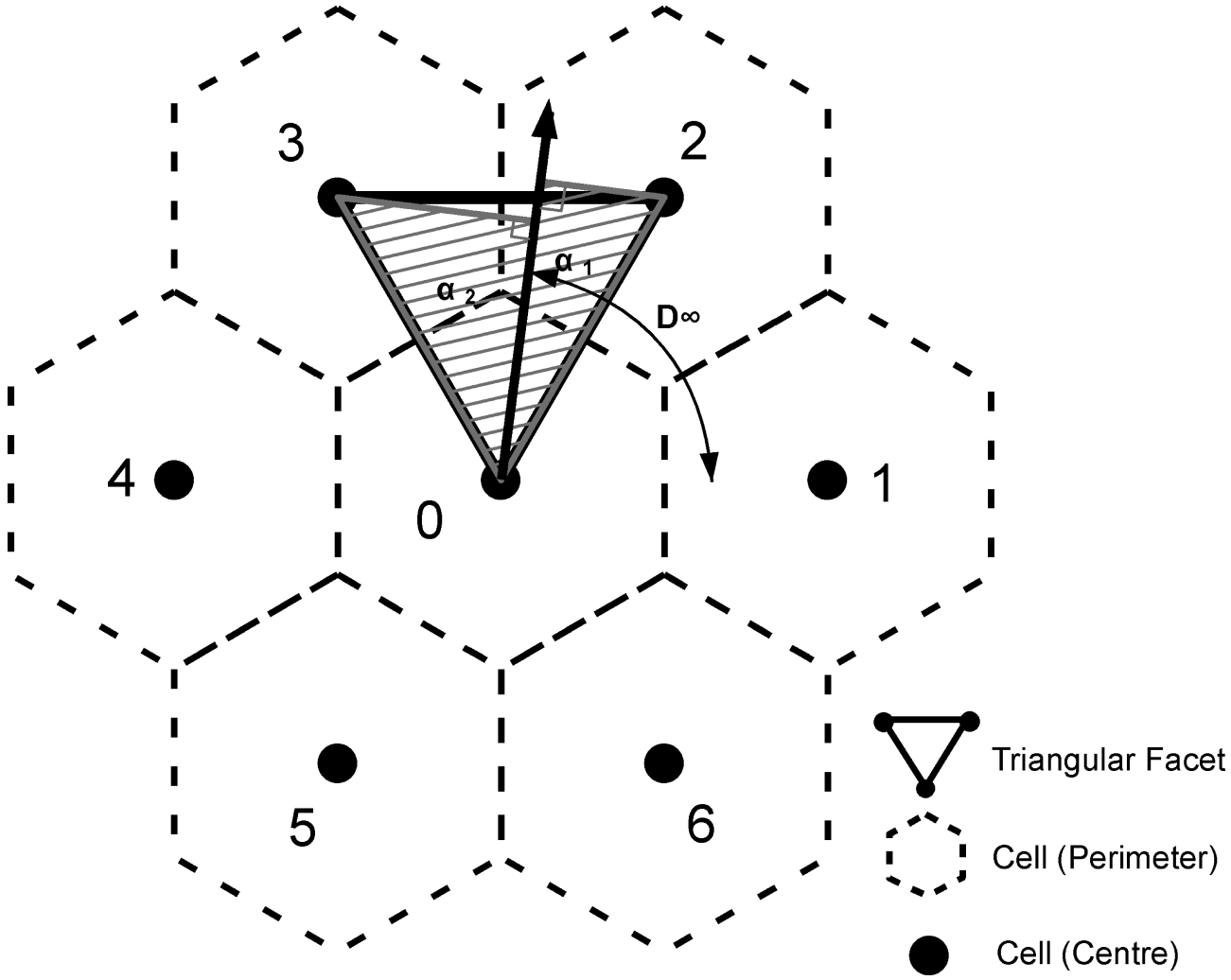


Figure 8.10.: Areal D∞ apportionment for RHSM-hex.

the contribution from a cell to the parent is apportioned for all the methods described above. Only one method for flow from fine to coarse was considered it is also shown in Figure 8.11.

Figure 8.12 shows the options considered for $D\infty$ flow accumulation when flow is from coarse to fine. In all the examples, a flow of 7 units is distributed; 4 and 3 to the coarse neighbours and then distributed amongst the recipients' children. "All coarse" distributes the flows to all the children of each neighbour. "Centre" distributes flow to only the centre child of the recipients. "Leading coarse" divides the flow received by each recipient evenly between the children that share part of their perimeter with the contributor. "Leading fine" distributes flow only to the child that is closest to the direction vector in each coarse neighbour. "Leading 4" distributes the flow to each neighbour proportional to the distance to the direction vector.

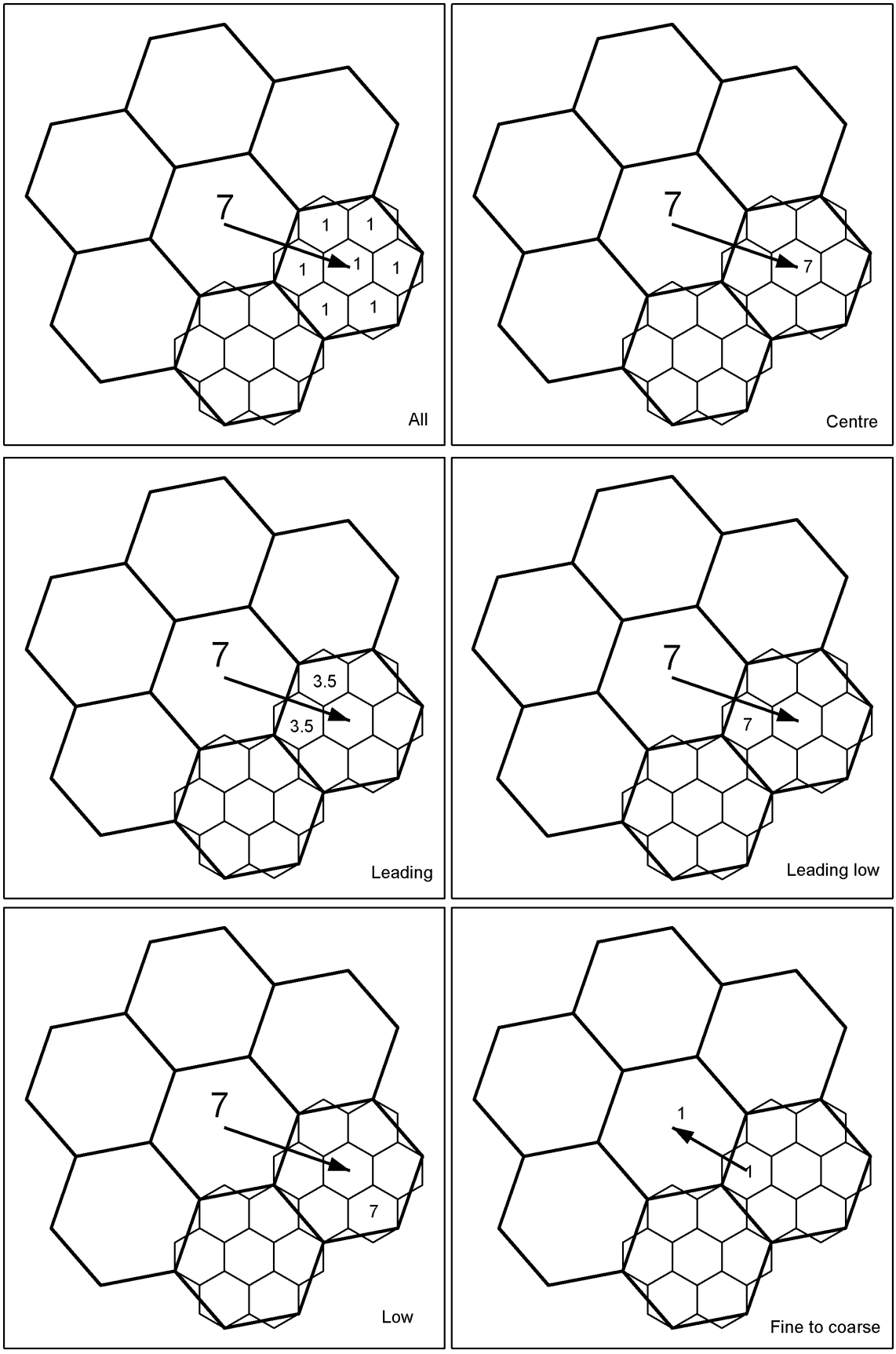


Figure 8.11.: *Dx* accumulation options in variable resolution for RHSM-hex.

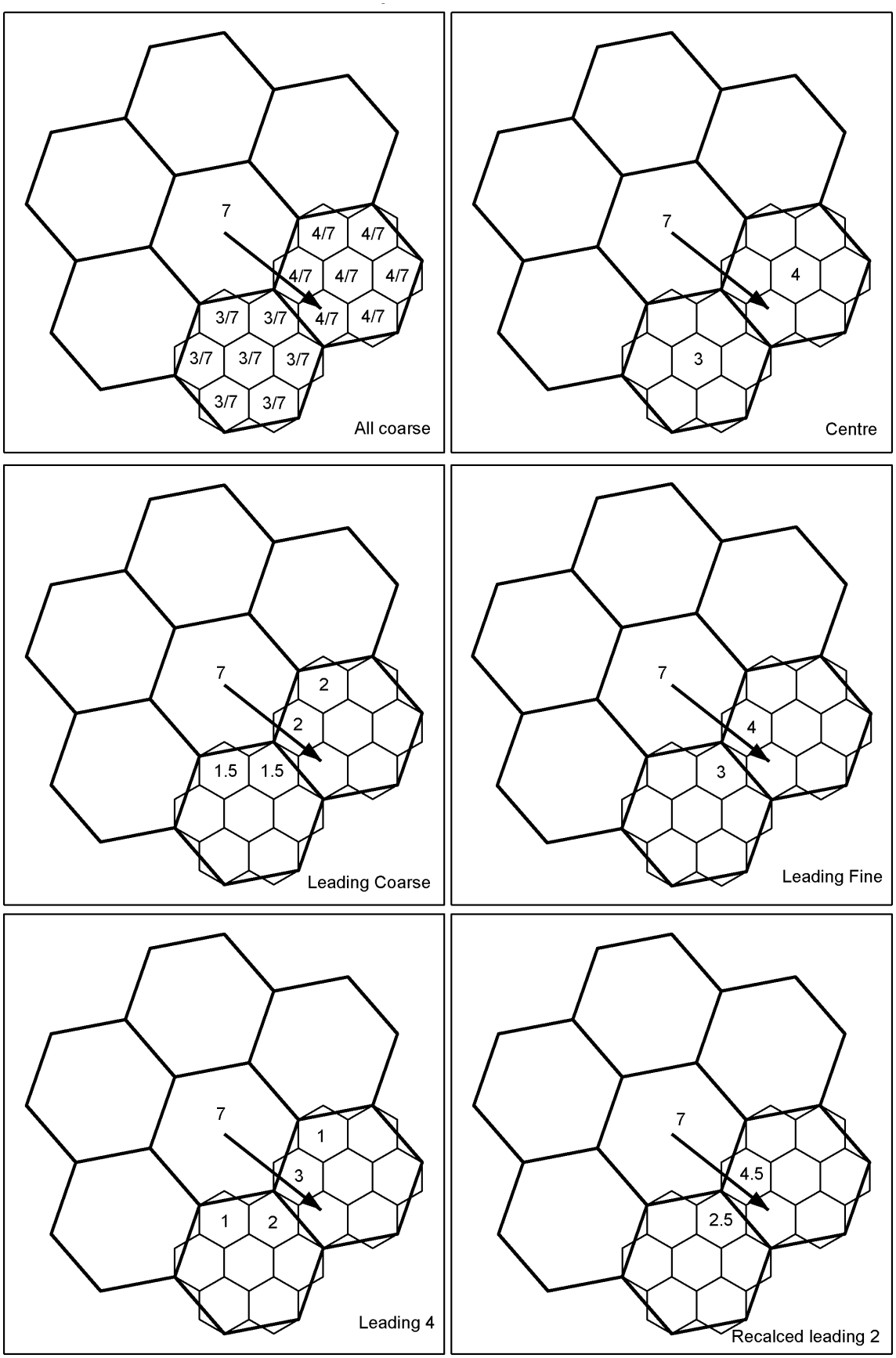


Figure 8.12.: D∞ accumulation options in variable resolution hexagonally sampled space.

All the above methods are consistent between Levels of Detail. “Recalced 2” recalculates the flow apportionment between the two closest recipient children. “Recalced 1” and “Recalced 4” are also possible but not shown. All the Recalced methods would potentially generate discrepancies between LoDs, which makes them unappealing. It is also possible to use a low method which would assign the flow to only the lowest elevation child in each coarse neighbour receiving flows. Leading coarse, Leading fine, and Leading four avoid both jumping flows and recalculation.

Figure 8.13 shows how flow is apportioned when flow is from fine to coarse. The flows are assigned to the parent of the cell it is apportioned to in the fine resolution. Three possible cases are shown. They are all scenarios of the same system.

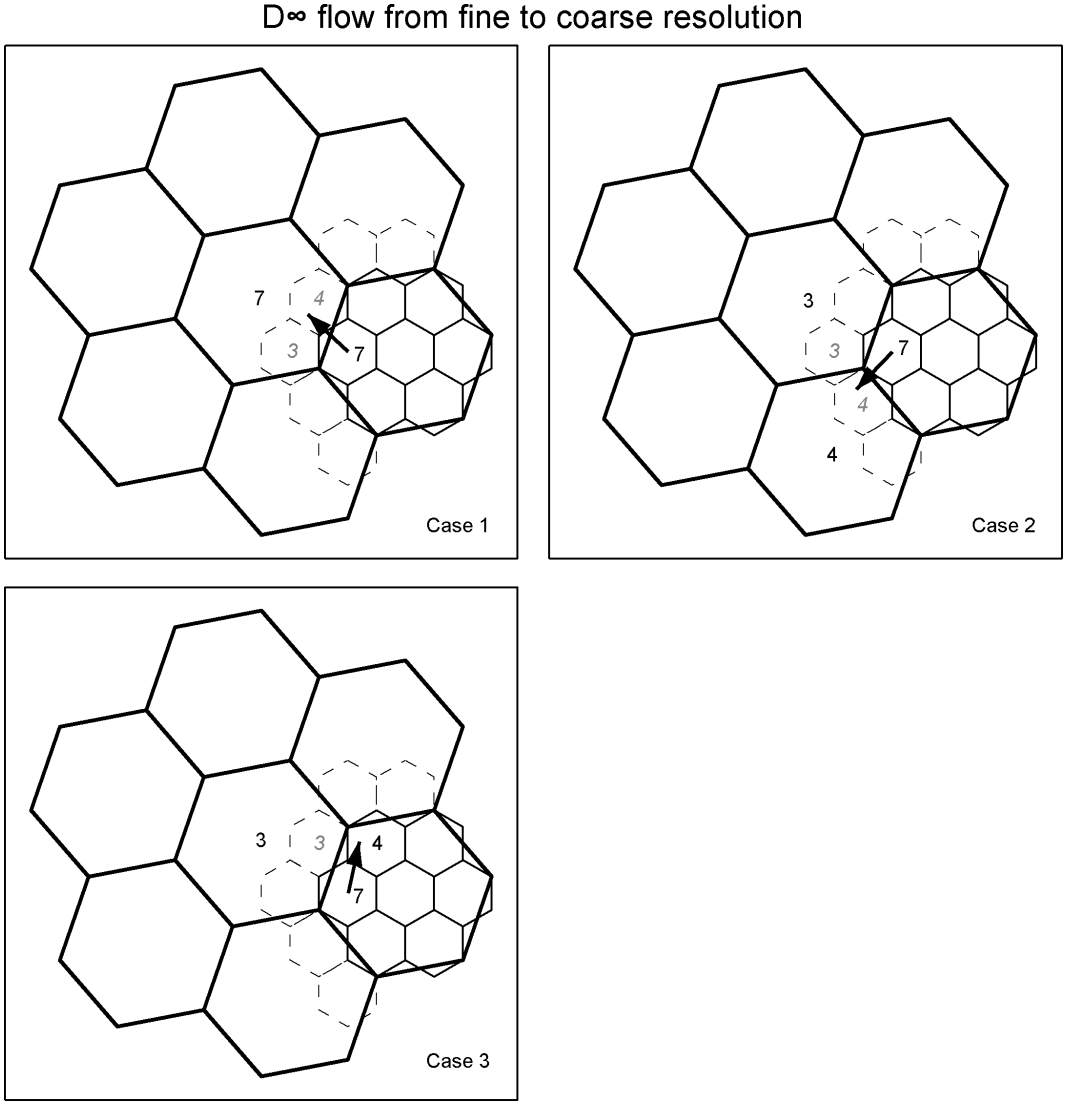


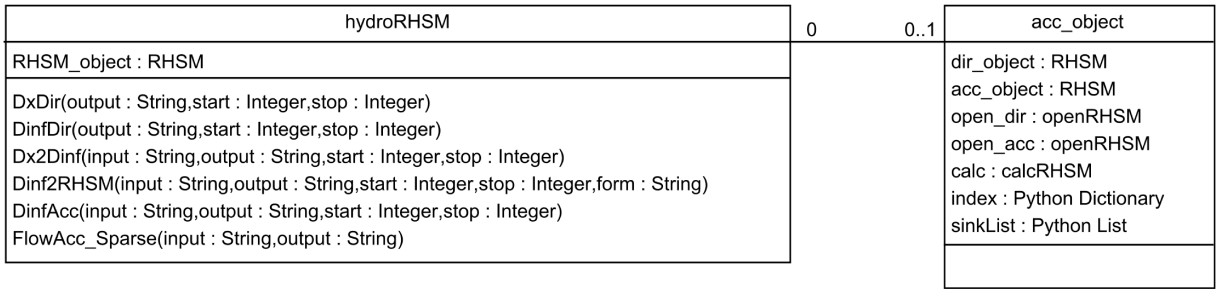
Figure 8.13.: D∞ flow from fine to coarse resolution in variable resolution hexagonally sampled space.

8.6. Hydro class specification

The hydrological modelling module was implemented using the class structure shown in Table 8.2. The hydrological modelling class is encoded separately from the main RHSM class to encourage modular development of new functions and applications, and to keep the main class relatively simple. The acc_object is encapsulated primarily to organise the parameters required for accumulation and to improve efficiency; it has no public methods.

DxDir determines Dx directions, DinfDir determines $D\infty$ directions, Dx2Dinf converts Dx into $D\infty$ radian directions. Dinf2RHSM converts $D\infty$ directions into RHSM directions, including implementation of the different ratio forms of $D\infty$ described in Subsection 8.5.2, DinfAcc calculates flow accumulations for $D\infty$ or Dx. FlowAcc_Sparse determines accumulations on a sparse array.

Table 8.2.: Hydro UML class specification



This chapter utilised the RHSM to create variable resolution flow direction arrays. It has also adapted flow direction and flow accumulation algorithms so that they are functional in the RHSM, primarily by generalising the algorithms so that they can apply to any of the three regular tessellations, rather than only the rectangular array on which they were originally conceived. In addition, a variation of the flow accumulation algorithm has been proposed to operate on a variable density realisation of an RHSM surface. The next chapter describes and evaluates a series of trials of the proposed algorithms.

9. Evaluation of RHSM for Hydrological Applications

This chapter evaluates the RHSM introduced in Part II for a specific task; that task is determining flow direction and flow accumulation arrays using the methods from hydrological geomorphometry literature, called $D8$ and $D\infty$, which were summarised in Chapter 4 and adapted for multi resolution surfaces in Chapter 8. The evaluation compares results derived using $D8$ and $D\infty$ with analytically derived results on divergent and convergent cones. The cone analysis is augmented by case studies on representative urban surfaces. The flow algorithms are also assessed by analysing how errors differ between tessellation geometries.

9.1. Overview

9.1.1. The experiments

In the following sections, the evaluation of the surface models contrasts hexagonal, rectangular, and triangular array sampling, as well as hierarchical and single LoD structures. The criteria for assessing the effectiveness of the RHSM, in this context, are to provide results that are isotropic, that is, they do not have directional bias and are consistent across scales whilst reducing the number of cells that are required for processing. Such capability would assist the capture of complex urban drainage patterns on commodity computers, including fine surface features with extensive effects.

Section 9.2 and Section 9.3 do not utilise the variable density or multi resolution capabilities of the RHSM, instead, they examine accuracy and anistropy of flow direction and accumulation respectively. The results serve as a baseline for the multi-resolution studies that follow in Section 9.4 and Section 9.5. Section 9.2 and Section 9.3 shed light on the performance of the algorithms by highlighting the differences in anisotropy found when comparing geometries. Section 9.3 also assesses and compares the three varieties of $D\infty$ proposed in Section 8.5 : Linear, Angular, and Areal.

Section 9.4 and Section 9.5 introduce multi-scale elements to the accuracy and anistropy analysis of flow direction and accumulation by creating variable resolution surfaces of flow direction arrays, calculating flow accumulation arrays from them and comparing the results to analytical values for a range of tolerances.

Section 9.6 applies the modelling process to real world urban surfaces from Dunedin, New Zealand. Compression ratios were measured for a range of tolerances. Given the newness and complexity of the methods developed and tested here, comprehensive analysis in urban areas is beyond the scope of this study. Section 9.7 summarises the results from all the assessments.

9.1.2. The mathematical surfaces

Not withstanding the real world case studies in Section 9.6, the trials in this chapter were undertaken on cone surfaces: concentrative and dispersive. Concentrative cones are formed with:

$$z = \sqrt{x^2/100^2 + y^2/100^2} \tag{9.1}$$

Dispersive cones are formed with:

$$z = 1.2 - \sqrt{x^2/100^2 + y^2/100^2} \tag{9.2}$$

In both cases, z is the elevation, and x, y are the Cartesian coordinates of the cell centre. The origin of the dataset is the centre of the cone. Equation 9.1 forms a downward pointing cone. The centre of the cone is the low point, therefore, the flow direction is toward the centre. Consequently, it is concentrative: flows concentrate as they descend (see Figure Figure 9.1 on page 200). Equation 9.2 forms an upward pointing cone. The centre of the cone is the high point, the flow direction is away from the centre of the cone. It is a dispersive cone: flows disperse as they descend.

The projection onto the (x, y) plane of both surfaces is limited to a radius 120 circle defined by:

$$x^2 + y^2 \leq 120^2 \tag{9.3}$$

The cones formed by Equations 9.1 and 9.2 have a gradient of only 1 in 100 and would seem fairly flat to a person standing on its surface. The units of measurement are immaterial as the surfaces are not tied to a real world projection, therefore, the dimensions are reported unit-less. All areas are projected areas i.e. they are the areas of the projection of the surfaces onto the (x, y) plane.

The cell sizes are adjusted so that all cells have an area of 1. The cell sizes (cs) for RHSM-rec, RHSM-hex, and RHSM-tri are given in the following equations. Cell size is defined as the centre to centre distances between adjacent cells as defined by the point set matrix $\mathbf{R}$ in Section 3.5.

$$cs_{rec} = 1 \tag{9.4}$$

$$cs_{hex} = \sqrt{2/\sqrt{3}} \tag{9.5}$$

$$cs_{tri} = \frac{4}{\sqrt{3\sqrt{3}}} \tag{9.6}$$

In order to fill the circle described in Equation 9.3 completely with this cell size, a minimum of the following number of levels are are required. RHSM-rec: 5, RHSM-hex: 6, and RHSM-tri: 9. The cardinality for RHSM-rec is 45255; for RHSM-hex, 45223; and for RHSM-tri, 45232. Only the cells within the circle defined by 9.3 are included in the cardinality counts.

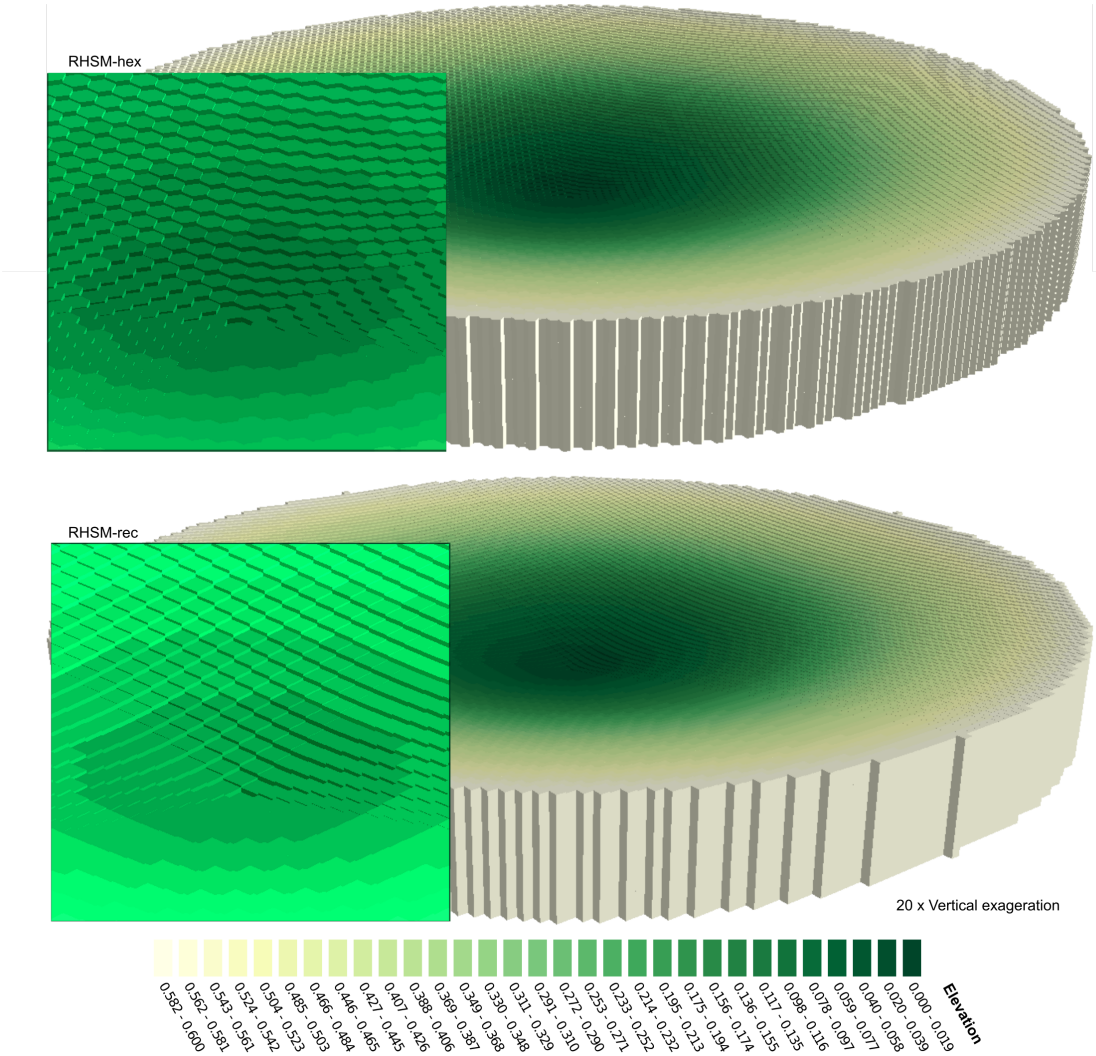


Figure 9.1.: : Perspective views of concentrative cone surface representations in RHSM-rec and RHSM-hex with near zoom of the centres inset.

The analytical solutions for flow direction were determined using:

$$\begin{aligned} \beta_{conc} &= \tan^{-1}\left(\left(\frac{-y}{-x}\right) + 2\pi\right) \%2\pi \\ \beta_{disp} &= \tan^{-1}\left(\left(\frac{y}{x}\right) + 2\pi\right) \%2\pi \end{aligned} \tag{9.7}$$

Where β_{conc} is the analytical direction on a concentrative cone, β_{disp} is the analytical direction on a dispersive cone, and x, y are the Cartesian coordinates of the point for which an analytical solution is determined. The modulus function (%) is applied to express the results as positive directions.

Gallant and Hutchinson (2011) provide the following to determine specific catchment area (a) for a convergent cone.

$$a_{conc} = \frac{l\,(2R - l)}{2\,(R - l)} \tag{9.8}$$

where l is the flow length and R is the radius of the cone for which accumulation is determined. The analytical solutions for Specific Catchment Area (a) were determined using:

$$\begin{aligned} a_{conc} &= \frac{\left(R - \sqrt{x^2+y^2}\right)\left(2R - \left(R - \sqrt{x^2+y^2}\right)\right)}{2\left(R - \left(R - \sqrt{x^2+y^2}\right)\right)} \\ a_{disp} &= \frac{\sqrt{x^2+y^2}}{2} \end{aligned} \tag{9.9}$$

where a_{conc} is the Specific Catchment Area on a concentrative cone, a_{disp} is the Specific Catchment Area on a dispersive cone, and x, y are the Cartesian coordinates of the point for which an analytical solution is determined. The flow direction of edge cells is undefined; therefore, they do not contribute to flow accumulation. Consequently, in order to provide an unbiased numerical comparison, $R = 119$ for a determined from cones limited by Equation 9.3.

The cone surface used in this analysis may not accurately assess how well variable resolution surfaces will capture flow networks in real world situations. Cone surfaces are highly structured, lacking both structural and random variation. However, cone surfaces have

two characteristics that make them an appropriate first hurdle for testing geomorphometric flow direction algorithms. Cone surfaces feature the full range of possible directions, which accentuates any inconsistencies in the algorithm due to directional biases inherent in the regular structures upon which tessellations are built. Furthermore, for a region of a given size, the rate of change in direction across that region decreases with distance from the cone centre. Therefore, applying an algorithm to a cone surface illustrates its response to a wide range of directions and rates of change in direction.

Real surface case studies were undertaken using a surface generated from first return LiDAR signals in areas identified as building footprints by photogrammetric analysis and ground return elsewhere.

9.1.3. Metrics for comparison

Relative Mean Absolute Error

A standard metric, which has been used in the literature for assessing the accuracy of flow direction and flow accumulation is Relative Mean Absolute Error (RMAE)[1] (Pan et al., 2004; Vázquez and Feyen, 2007), a practise that will be followed here. However, RMAE does not capture positional errors, only attribute errors, and position errors are particularly important for flow accumulation on real world surfaces.

RMAE can only be calculated where results accepted as truth are known to the experimenter. This definition of truth is provided for flow accumulation and direction on the mathematical surfaces where flow accumulation, or more precisely specific catchment area (a), can be determined analytically from first principles. For real world case studies, analytical results are not generally available.

However, accuracy of spatial variables cannot be determined by comparison with analytical point results or measurements unless there is agreement between scales. Many variables that vary with scale can never be precisely known as a closer examination will reveal imprecision in the coarser measure, not because the new measure is merely more precise but because the scaling function of the variable being evaluated is non-linear and unknown.[2]

The form of the equations used by Pan et al. (2004), who were primarily concerned with TWI, can be simplified for flow accumulation and direction. The error (E) of the of cell

[1] Also called relative root mean square error.

[2] Some authors recommend using multiple resolutions to analyse spatially variable outputs. For instance, Shary et al. (2002) suggest finding outputs that tend to a limit as resolution increases, and, in a similar fashion, Florinsky and Kuryakova (2000) search for smooth areas in graphs that plot model correlation against resolution.

i is the difference between the computed (a_{Ci}) and theoretical (a_{Ti}) values .

$$E_i = a_{Ci} - a_{Ti} \tag{9.10}$$

To investigate the underlying cause of anistropy the sign of the error may be important. Flow accumulation values vary greatly within a dataset with some cells having values many orders of magnitude higher, therefore, we want to report relative error (RE) for each cell using:

$$RE_i = \frac{a_{Ci} - a_{Ti}}{a_{Ti}} \tag{9.11}$$

An unsigned summary value for the entire dataset, Mean Absolute Error (MAE), can be found using:

$$MAE = \sqrt{\sum_{i=1}^{i=n} \frac{(E_i)^2}{n}} \tag{9.12}$$

Alternately, Relative Mean Absolute Error ($RMAE$) can be found by substituting RE_i for E_i in Equation 9.12.

RHSM Angular units

Some anistropy results below are plotted in angular RHSM units rather than degrees or radians. One RHSM unit is the angle subtended between the cell centres of adjacent neighbours. One RHSM angle unit is equivalent to $\pi/4$, $\pi/3$ or $2\pi/3$ for RHSM-rec, RHSM-hex, or RHSM-tri respectively (see Figure 8.2 on page 168). Normalising anistropy results by RHSM angle facilitates comparison of algorithms and geometries because one RHSM unit represents the range of one $D8$ direction or one $D\infty$ facet.

9.2. Cone direction Dx and $D\infty$ direction accuracy and anistropy

9.2.1. Overview

The aim of this experiment was to quantify the $RMAE$ of Dx and $D\infty$ flow direction algorithms in RHSM-rec, RHSM-tri, and RHSM-hex by comparing flow direction arrays with solutions derived analytically on dispersive and concentrative cones. In addition, the anisotropy of the Dx and $D\infty$ flow direction algorithms were explored by mapping the spatial distribution of the RE. Comparing the anistropy of the different geometries provides insight into the nature of the geometric bias inherent in the algorithms. $D\infty$ is more accurate than Dx, however, there are still errors and geometric bias in the direction.

9.2.2. Methods

The experiment uses the dispersive and concentrative cone surfaces described in Subsection 9.1.2. Using RHSM-rec, RHSM-tri, and RHSM-hex tessellations, flow direction arrays were defined using $D\infty$, Dx and analytically. The analytical solutions were determined using Equation 9.7. The resulting directions are expressed in radians anti-clockwise from the x-axis.

The analytical solutions were considered to be correct and were subtracted from the other datasets determined on equivalent tessellations to produce error (E) arrays using Equation 9.10. The error array was also plotted against analytical direction to highlight the anistropy.

The error of each dataset was summarised by a Mean Absolute Error (MAE). The central cell was excluded from the MAE because, given that all directions have an equal slope, it assumes an arbitrary direction based on the order in which the algorithm considers neighbours.

9.2.3. Results

The flow direction and error arrays for the concentrative cone are shown in Figure 9.2. The spatial distributions of errors for the dispersive cone are shown in Figure 9.3. Directions are shown in red and errors are shown in diverging colours; blue is positive error (clockwise of analytical), brown is negative error (anticlockwise of analytical), lighter colours indicate less error.

The RHSM-tri results are poorly represented at this scale so a near zoom of part of the result is shown in Figure 9.4. The RHSM-tri creates an alternating pattern which at far zoom appears similar to RHSM-hex because the sum of two consecutive vectors of RHSM-tri is equivalent to an appropriately scaled RHSM-hex vector (Figure 9.5).

The MAEs are summarised in Figure 9.6. $D\infty$ directions are roughly 2 orders of magnitude more accurate across all RHSM geometries. In both algorithms, rectangular is more accurate than hexagonal and both are notably more accurate than triangular. There is no difference in the MAE between dispersive and concentrative cones using Dx algorithm. However, $D\infty$ slightly more accurate in concentrative cones than in dispersive.

Dx errors are plotted against the analytical angle in RHSM units in Figure 9.7. The clearly defined lines in Figure 9.7 show that Dx results are invariant with elevation contour. The Dx results vary with direction with discontinuities at RHSM = 0.5. Dx errors are greatest at the divisions between the restricted directions and error free at the midpoint of each direction. RHSM-tri results form two lines due to the alternating neighbourhood directions.

$D\infty$ errors are plotted against the analytical angle in RHSM units in Figure 9.8. $D\infty$ errors vary with direction and have discontinuities where RHSM = 1 and 0, and least error where RHSM = 0.5. Locations without error in $D\infty$ results coincide with the greatest errors in Dx. In addition, Figure 9.9, which plots error against angle and distance from the cone centre, shows that $D\infty$ errors increase toward the centre of the cone.

There are thin strips of zero error where the analytical direction corresponds with the edge of a $D\infty$ facet edge. However, the error associated with contour is greatest in the cells adjacent these strips. The error then diminishes toward the centre of each facet. The sign of the error reverses between divergent and convergent cones. $D\infty$ is slightly more accurate in concentrative cones than it is in dispersive cones.

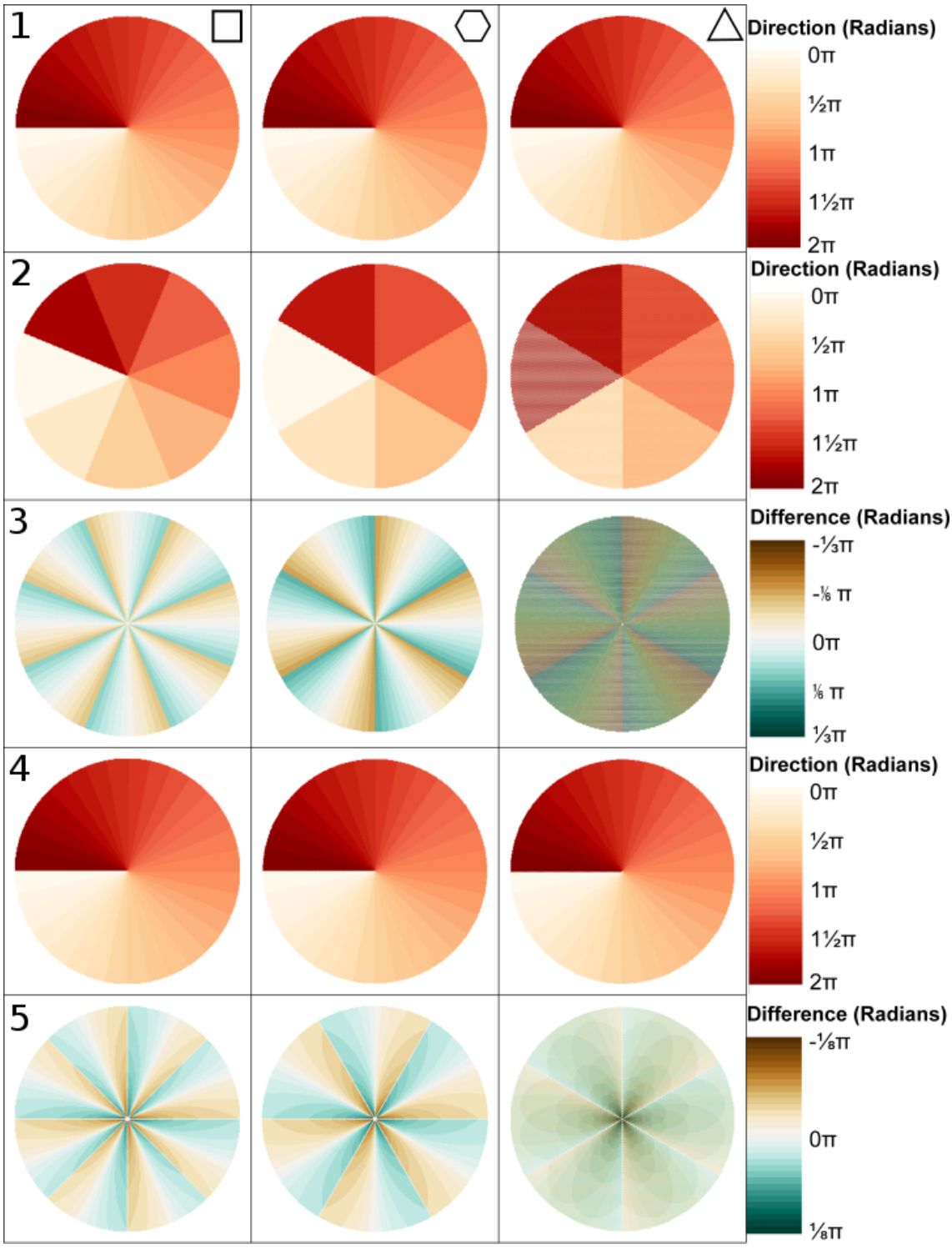


Figure 9.2.: Concentrative cone flow direction anistropy. The geometry of each column is indicated by the shape in Row 1. Row content is as follows; Row 1: Analytical, Row 2: Dx, Row 3: Analytical - Dx, Row 4: $D\infty$, Row 5: Analytical - $D\infty$. The Row 5 colour scheme has a smaller range than Row 3 in order to contrast the spatial distribution of error. See Figure 9.6 to contrast error magnitude between methods.

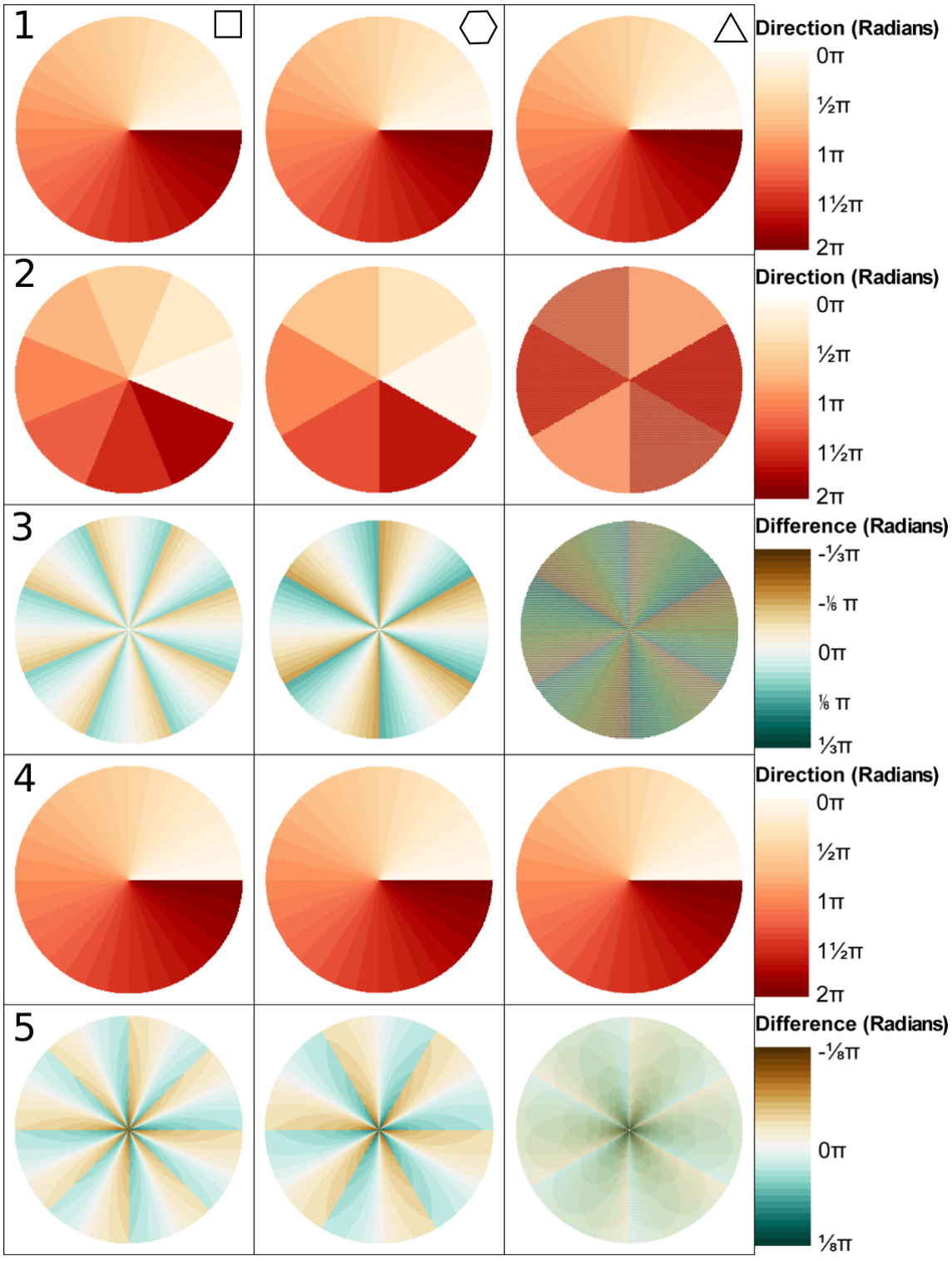


Figure 9.3.: Dispersive cone flow direction anistropy. The geometry of each column is indicated by the shape in Row 1. Row content is as follows; Row 1: Analytical, Row 2: Dx, Row 3: Analytical - Dx, Row 4: $D\infty$, Row 5: Analytical - $D\infty$. The Row 5 colour scheme has a smaller range than Row 3 in order to contrast the spatial distribution of error. See Figure 9.6 to contrast error magnitude between methods.

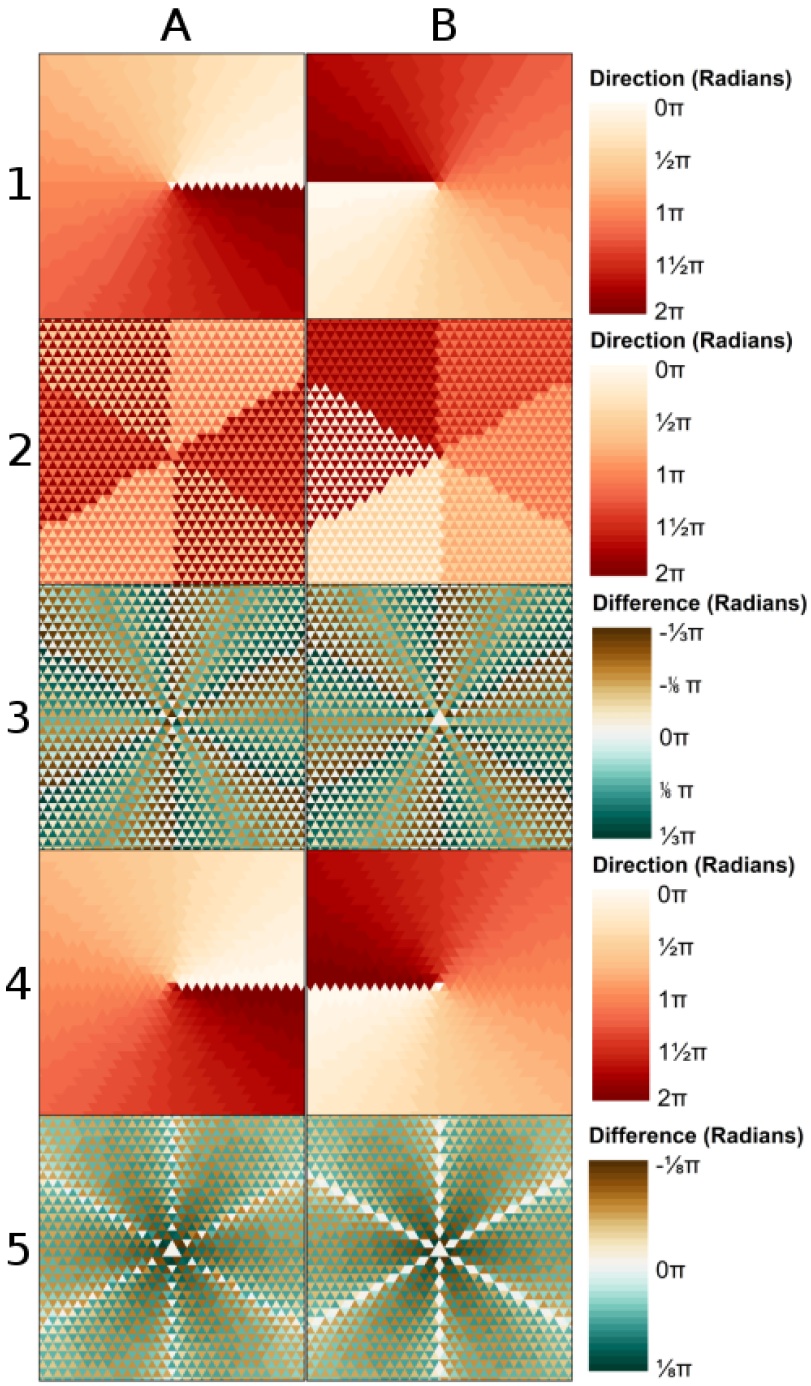


Figure 9.4.: RHSM-tri flow direction anistropy close up. Column A is a near zoom of the dispersive RHSM-tri directions in Figure 9.3, Column B is a near zoom of the concentrative RHSM-tri directions in Figure 9.2. Row content is as follows; Row 1: Analytical, Row 2: Dx, Row 3: Analytical - Dx, Row 4: $D\infty$, Row 5: Analytical - $D\infty$. The Row 5 colour scheme has a smaller range than Row 3 in order to contrast the spatial distribution of error. See Figure 9.6 to contrast error magnitude between methods.

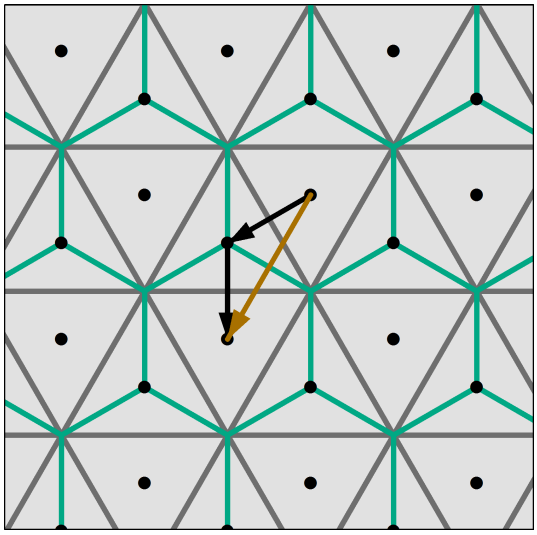


Figure 9.5.: Paths between hexagon and triangle cell centres. The path through three triangles shown in black is equivalent to the path between two hexagons shown in brown. Therefore, hexagonal and triangular flow directions form similar patterns.

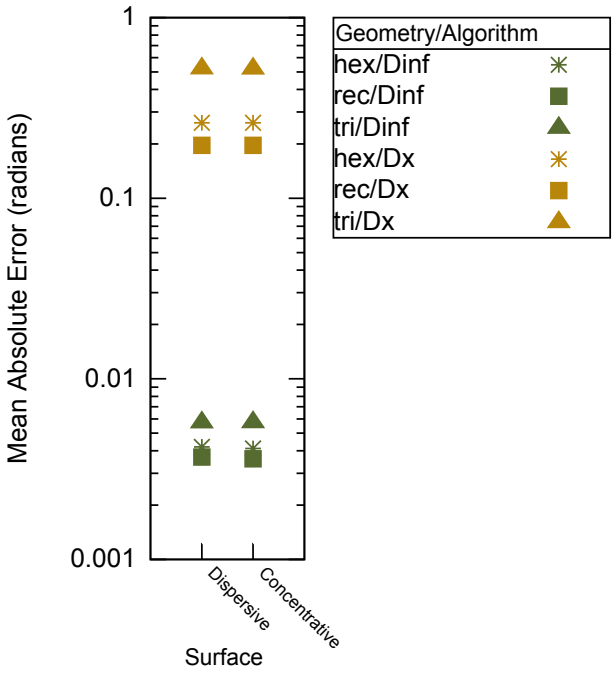


Figure 9.6.: Mean Absolute Error of direction algorithms. This graph compares the result of Dx and $D\infty$ algorithms to analytical results on dispersive and concentrative cones.

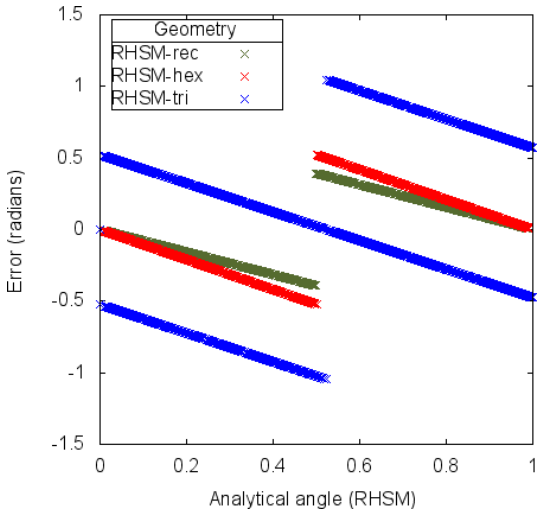


Figure 9.7.: Dispersive cone Dx flow direction anistropy. Errors are shown for the three geometries plotted against RHSM angle.

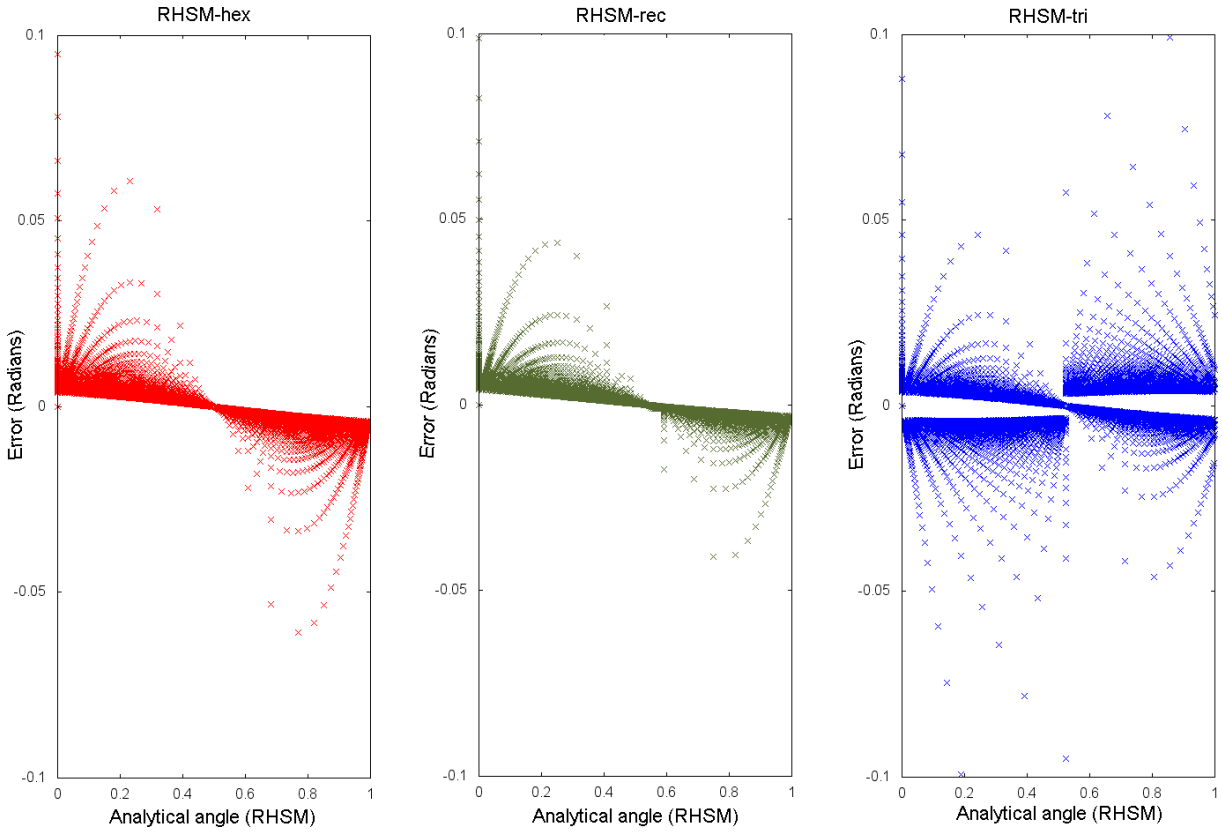


Figure 9.8.: Dispersive cone D∞ flow direction anistropy. Errors are shown for the three geometries plotted against RHSM angle.

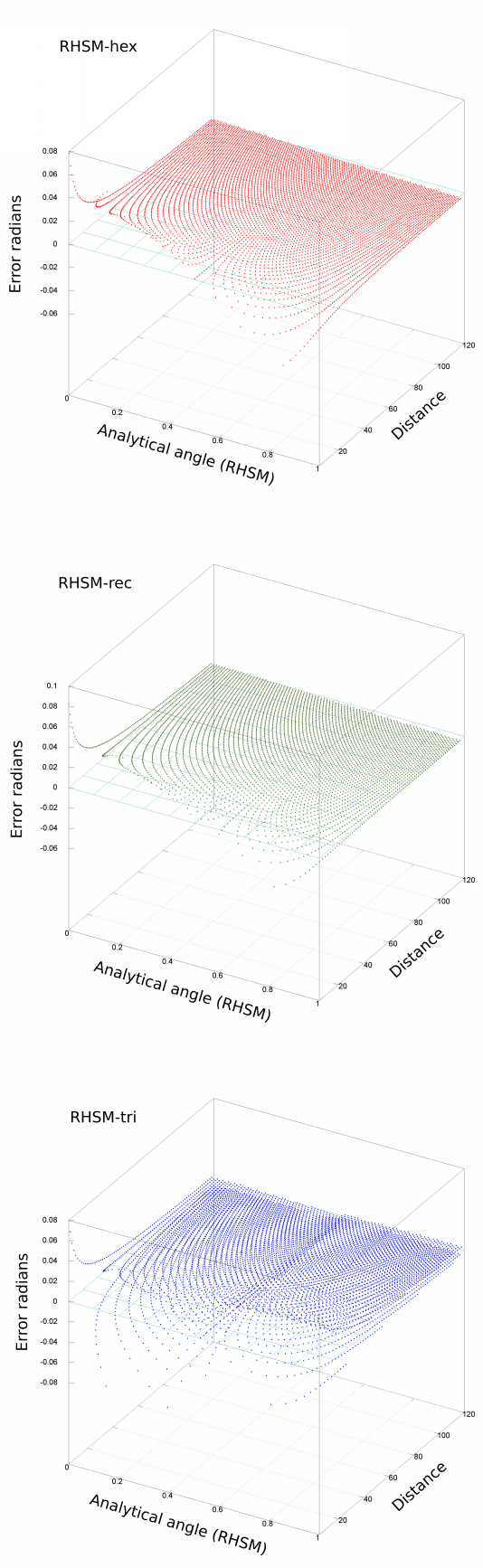


Figure 9.9.: Dispersive cone $D\infty$ direction anistropy by angle and distance to the cone centre.

9.2.4. Discussion

It is unsurprising that $D\infty$ is significantly more accurate than Dx. Dx directions are limited to the directions of neighbours, whereas, $D\infty$ varies continuously, which gives it the flexibility to model flow direction more accurately. The Dx accuracy of the different geometries of tessellation is related to the magnitude of angle subtended between neighbouring cell centres, which is also the angle that in Dx is simplified to a single direction. The greater the angle, the greater the maximum and mean error. In $D\infty$, the sign of the errors are reversed when comparing dispersive and concentrative cone in Figures 9.2 and 9.3 because reversing the direction of flow causes the facet to be formed with different nodes.

The difference in accuracy between the different geometries is very pronounced in Dx. However, the same ranking of geometries for accuracy is exhibited to a reduced extent in $D\infty$, which is due to the assumption that the aspect of the steepest facet applies to the entire cell, when it has, in fact, been calculated from only a single facet. On a cone surface, the direction of the steepest facet will be slightly more or less than the analytically determined direction of the cell, unless the steepest facet is directly in line with the cell centre and cone centre, which occurs only when the direction is in the centre of the facet; a situation that occurs when the direction is halfway between neighbours. The bias in $D\infty$ from the facet direction increased toward the centre of the cone because the range of angles represented by a single cell is greater, therefore, the potential for bias due to the non-alignment of the steepest facet is greater.

One possible explanation for why the dispersive errors are slightly larger than the concentrative errors is due to ambiguity in dispersive directions, which is discussed further in Section 9.3.

9.3. Cone accumulation Dx and $D\infty$ anistropy

9.3.1. Overview

This experiment determines the distribution of relative error of flow accumulation for a single LoD. Concentrative and dispersive cones are analysed for $D\infty$ and Dx. The experiment compares the three geometries for flow accumulations derived from Dx arrays but only RHSM-rec and RHSM-hex for $D\infty$ because the $D\infty$ algorithm does not function in triangular geometries. Three forms of $D\infty$ accumulation are examined (Linear, Angular, and Areal). $RMAE$s are calculated to rank the algorithms. The anistropy is

analysed by plotting relative error in 2-D to highlight algorithm artefacts. Due to the ordered structure of the cone surfaces, anisotropy is accentuated. The anistropy of the different geometries indicates that it is unhelpful to consider the cross-section of the cell in estimation of contour length for specific catchment area.

9.3.2. Methods

The flow accumulation algorithms Dx and the three versions of $D\infty$ were applied to the flow direction arrays generated in Section 9.2. Relative Error was determined by comparison with a for the cell centre. a was calculated using Equation 9.9. All cells have an area of 1. Therefore, the flow accumulation value was not adjusted to allow for the variation of cell size. The analytical angle of each cell is found by the same formula used to determine the analytical angle to test flow direction (Equation 9.7).

Relative Error arrays were mapped as colour intensity in 2-D, and also plotted against angle measured in RHSM units. Each Relative Error array was summarised by finding a trimmed $RMAE$. As cells get closer to the centre of the cone, at some point, the radius of curvature relative to cell size becomes significant and the relationship between flow accumulation and a breaks down. For this reason, a section of the cone near the centre, where the radius of curvature is greatest, is omitted from the statistics. This was accomplished by excluding the 5% most extreme error values.

Relative errors were also predicted for options 2 and 4 discussed in Subsection 8.5.1 to allow comparison with observed errors. The Relative Errors from cell width were modelled using the equations below.

$$RE_{2rec} = |\sin(D\infty)| + |\cos(D\infty)| - 1 \tag{9.13}$$

$$RE_{4rec} = \frac{1}{RE_{2rec}} - 1 \tag{9.14}$$

$$RE_{2hex} = \frac{\cos((D\infty + \pi/2)\,\%\pi/3 - \pi/6)}{\cos(\pi/6)} - 1 \tag{9.15}$$

$$RE_{4hex} = \frac{\cos\left(\pi/6\right)}{\cos\left(\left(D\infty\right)\%\pi/3 - \pi/6\right)} - 1 \tag{9.16}$$

Where RE_{2rec} and RE_{4rec} are the modelled errors for RHSM-rec using options 2 and 4 respectively, and RE_{2hex} and RE_{4hex} are the modelled errors for RHSM-hex.

9.3.3. Results

Dx Accumulations

The results are summarised as $RMAE$s in Figure 9.10. Dispersive cone errors are noticeably greater. The ranking of the geometries differs between the dispersive (rec, hex, tri) and concentrative (hex, tri, rec) cones. The Dx accumulations on the dispersive cone and concentrative cone are shown in Figures 9.11 and 9.12 respectively. A close up of an individual cell neighbourhood is shown under the results, this helps compare the anistropy of the results with the geometry of tessellation.

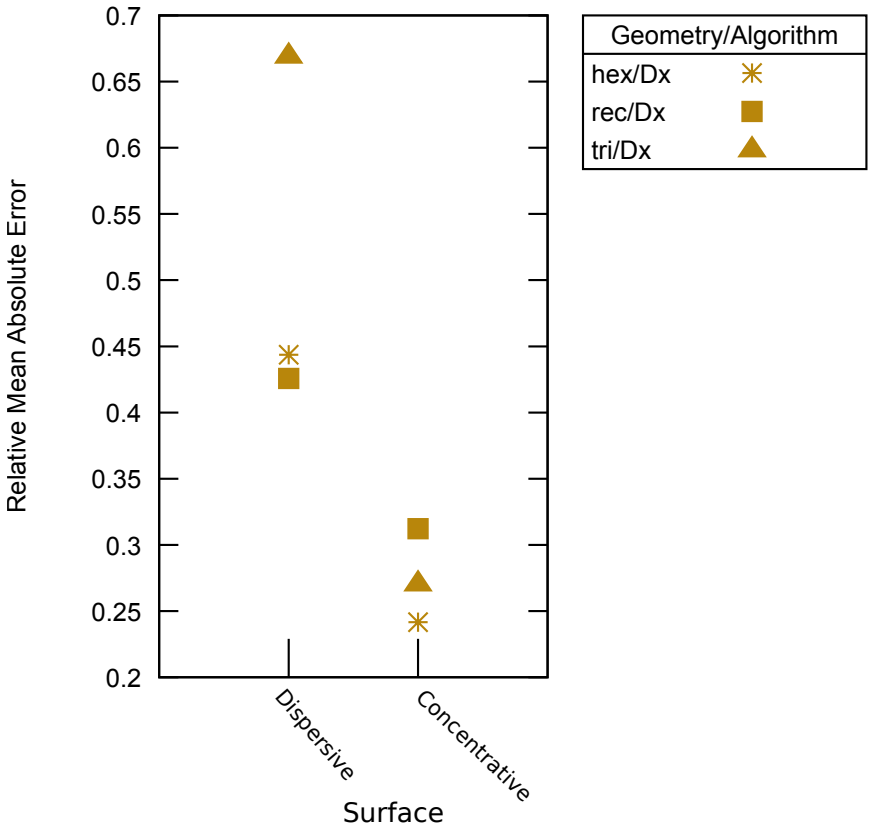


Figure 9.10.: Dx Accumulation summary

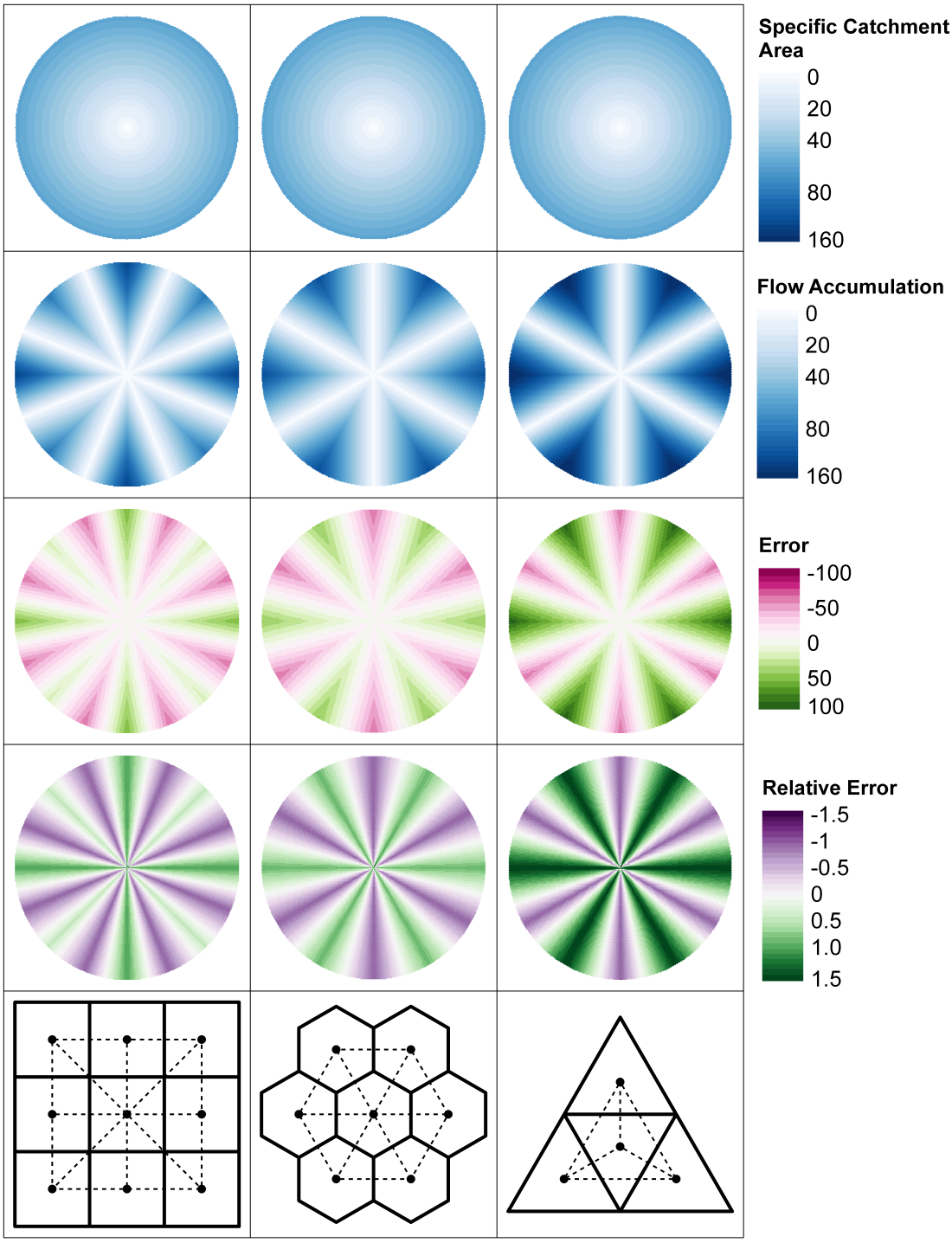


Figure 9.11.: Dx dispersive cone accumulations. Each column features the geometry in the bottom row.

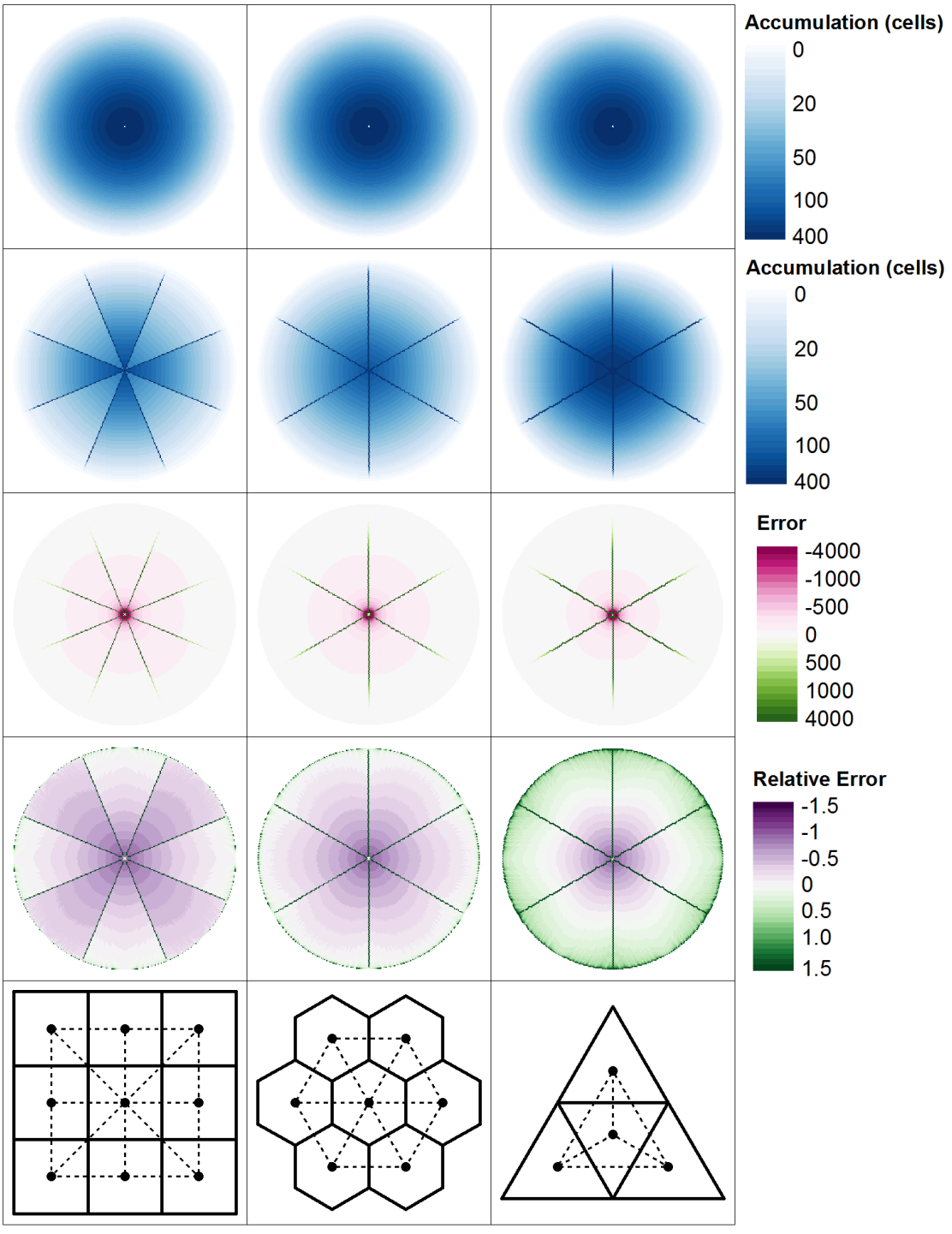


Figure 9.12.: *Dx* concentrative cone accumulations. Each column features the geometry in the bottom row.

D∞ Accumulations

The results for $D\infty$ Linear dispersive cone are shown in Figure 9.13, for $D\infty$ Angular dispersive cone in Figure 9.14, and $D\infty$ Areal dispersive cone in Figure 9.15. The equivalent results for concentrative cones are shown in Figure 9.16, Figure 9.17, and Figure 9.18 respectively. Accumulations are blue, error in light green and pink diverging, and Relative Error in dark green and purple diverging. A close up of an individual cell neighbourhood is shown above the results.

RHSM-hex accumulations are symmetrical across the facet; whereas, RHSM-rec accumulations are symmetrical only across adjacent pairs of facets, which corresponds to the neighbourhood symmetry. The exception is RHSM-rec angular dispersive, which is only rotationally symmetric.

D∞ Accumulations Dispersive

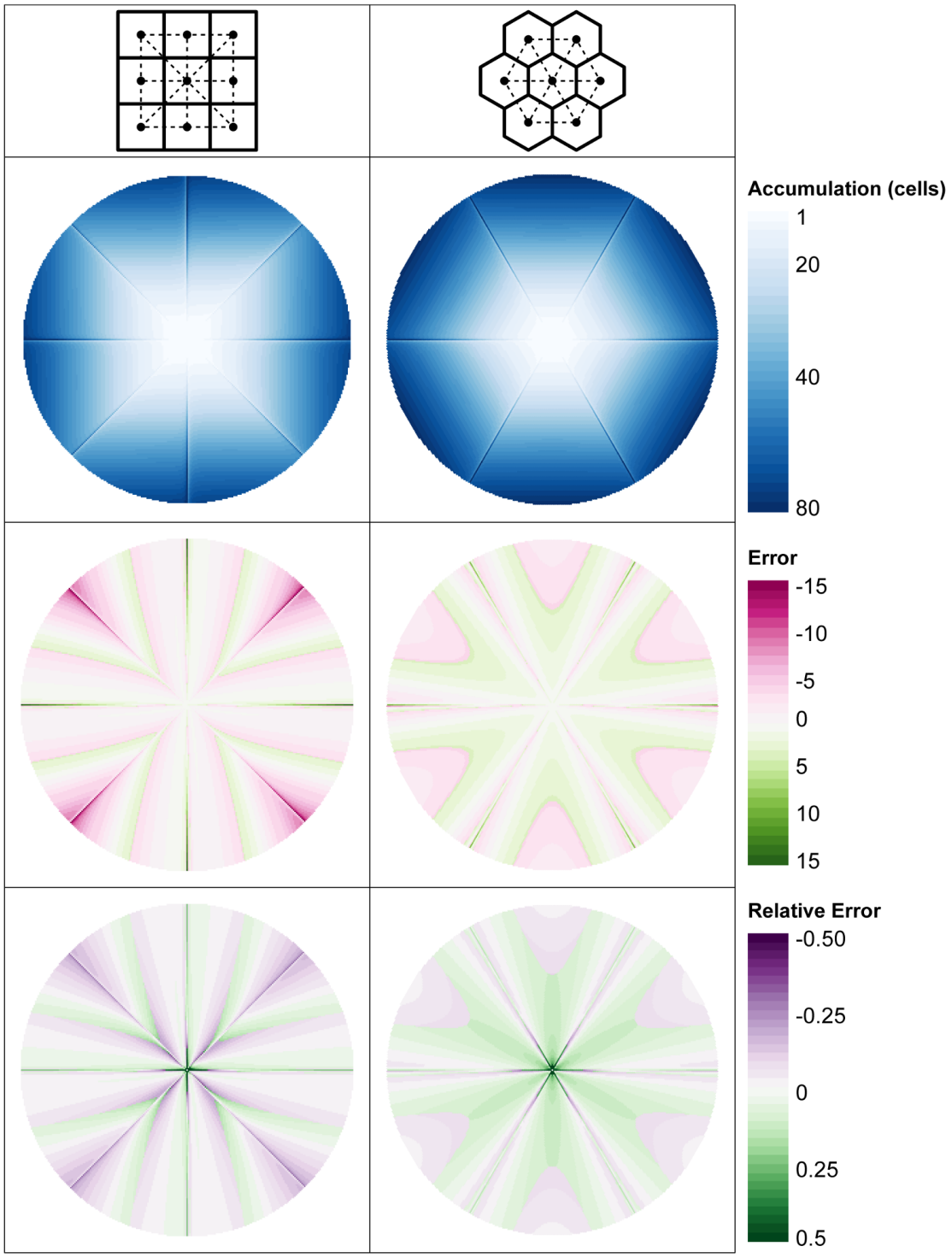


Figure 9.13.: $D\infty$ Linear accumulation on dispersive cones.

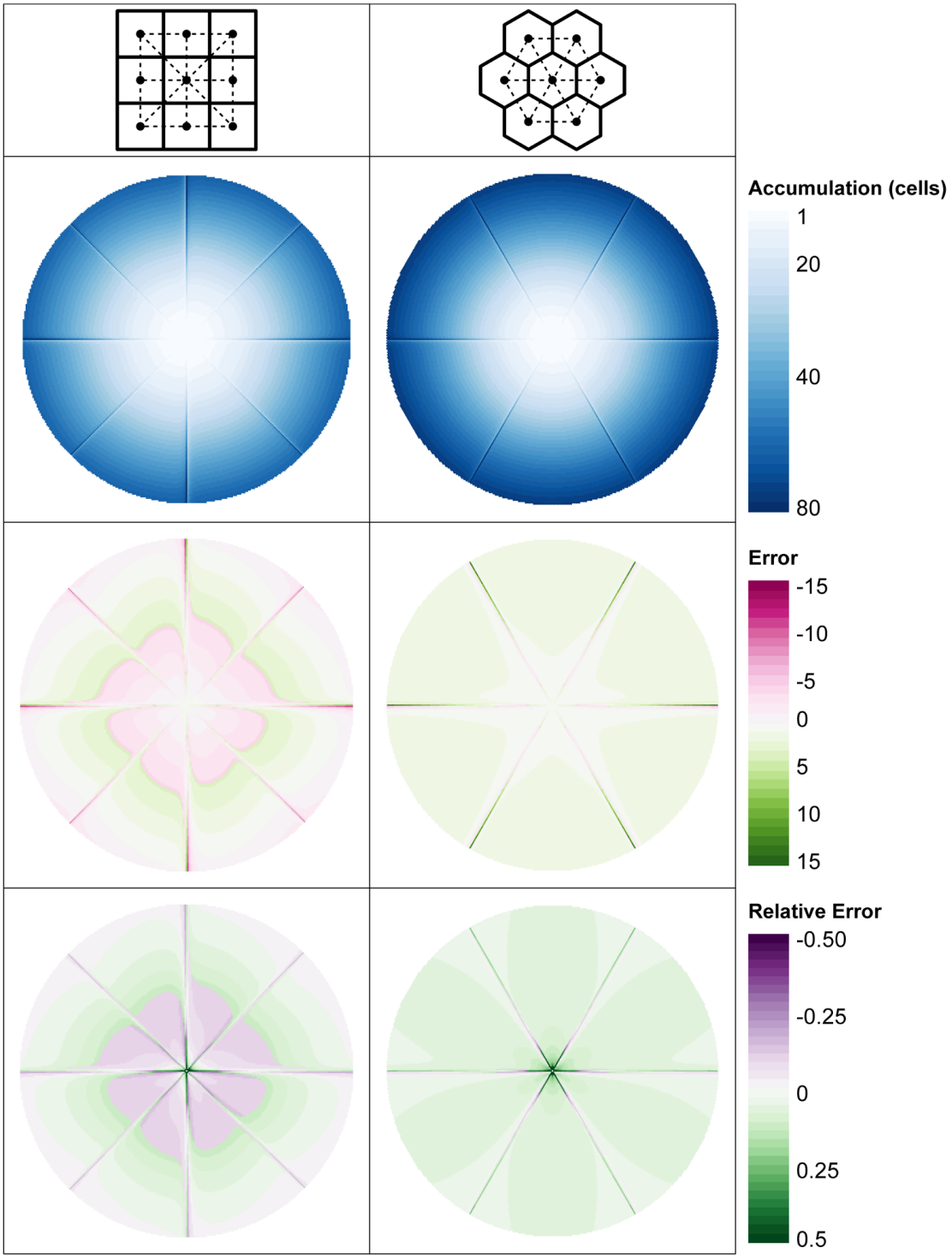


Figure 9.14.: $D\infty$ Angular accumulation on dispersive cones.

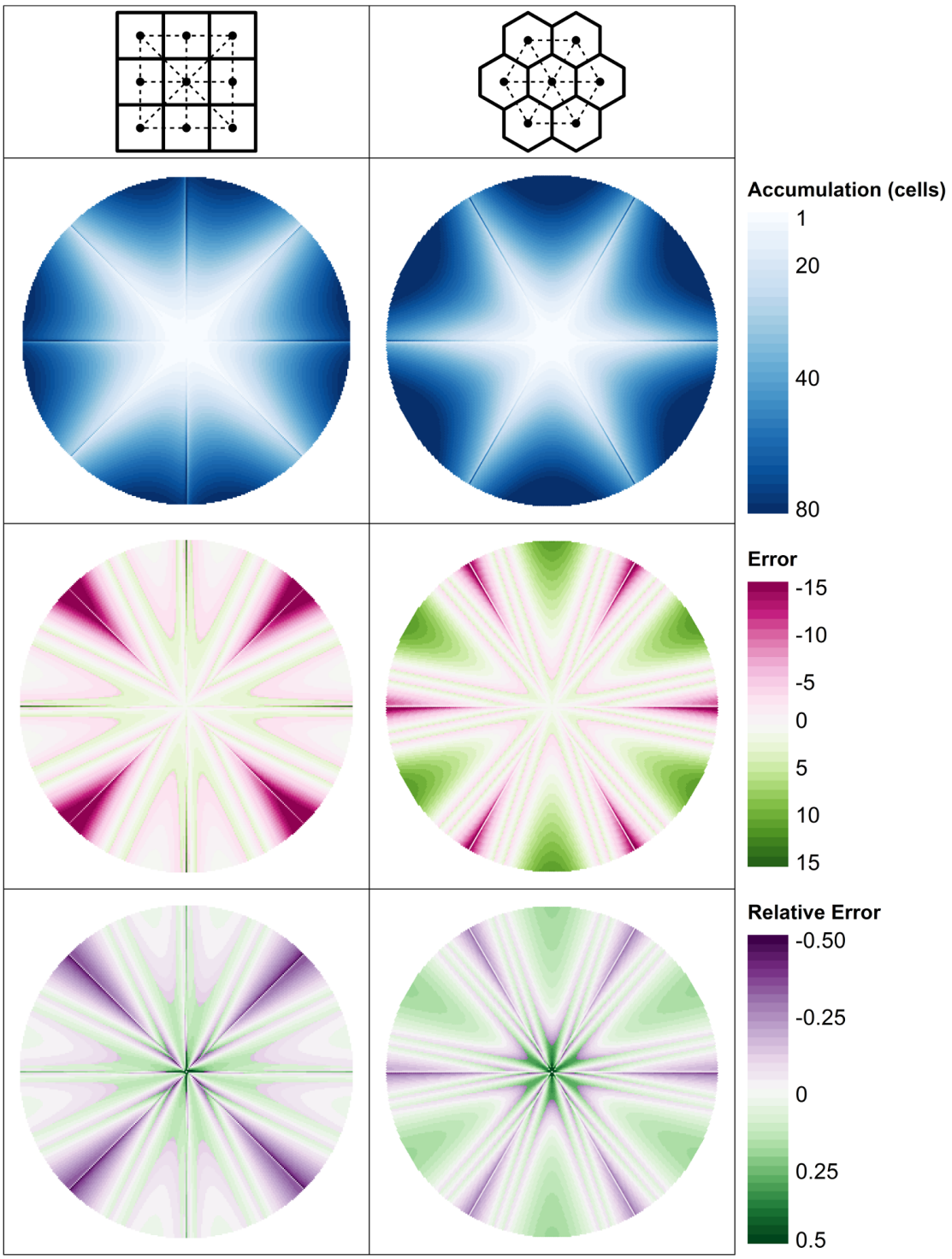


Figure 9.15.: $D\infty$ Areal accumulation on dispersive cones.

D∞ Accumulations Concentrative

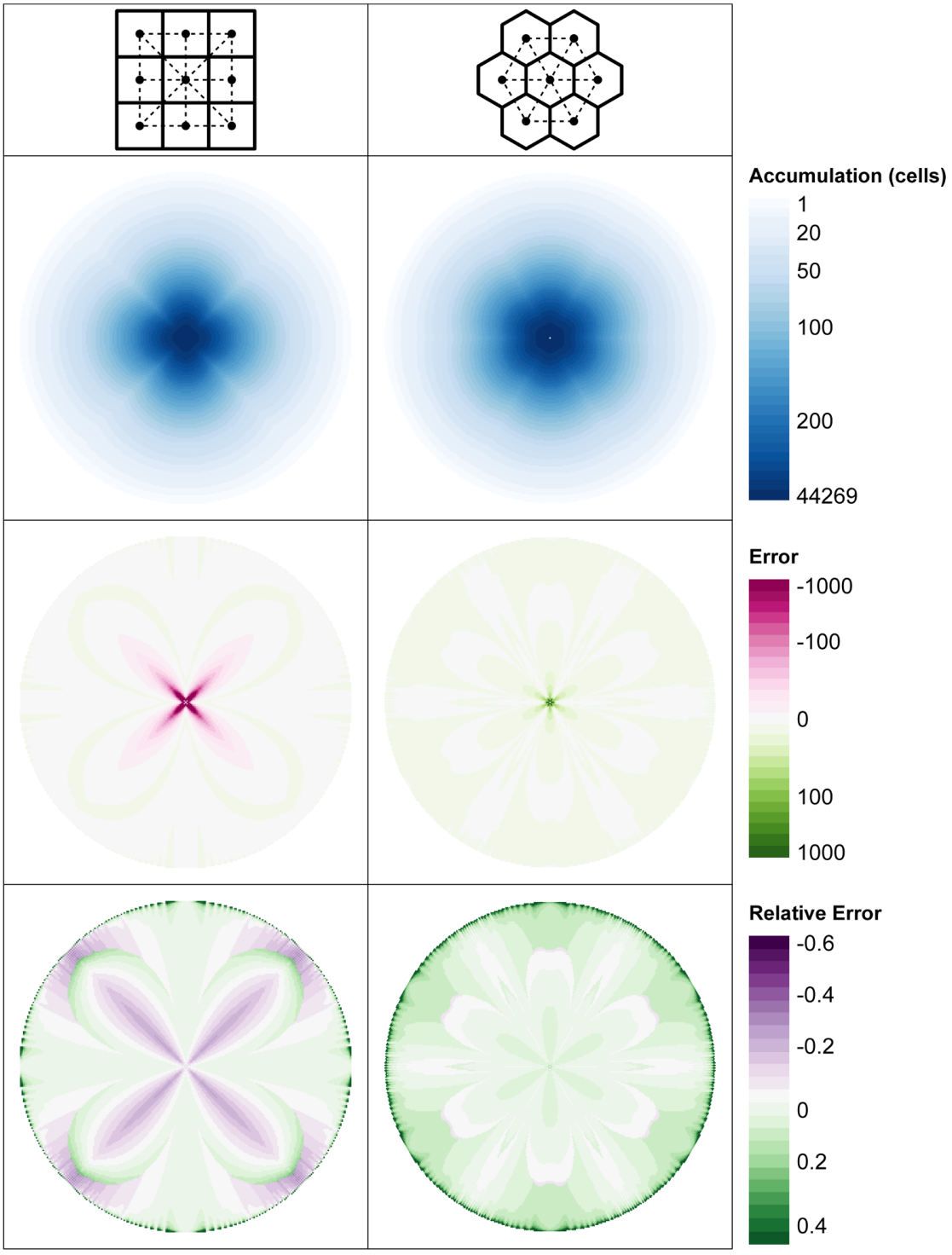


Figure 9.16.: $D\infty$ Linear accumulation on concentrative cones.

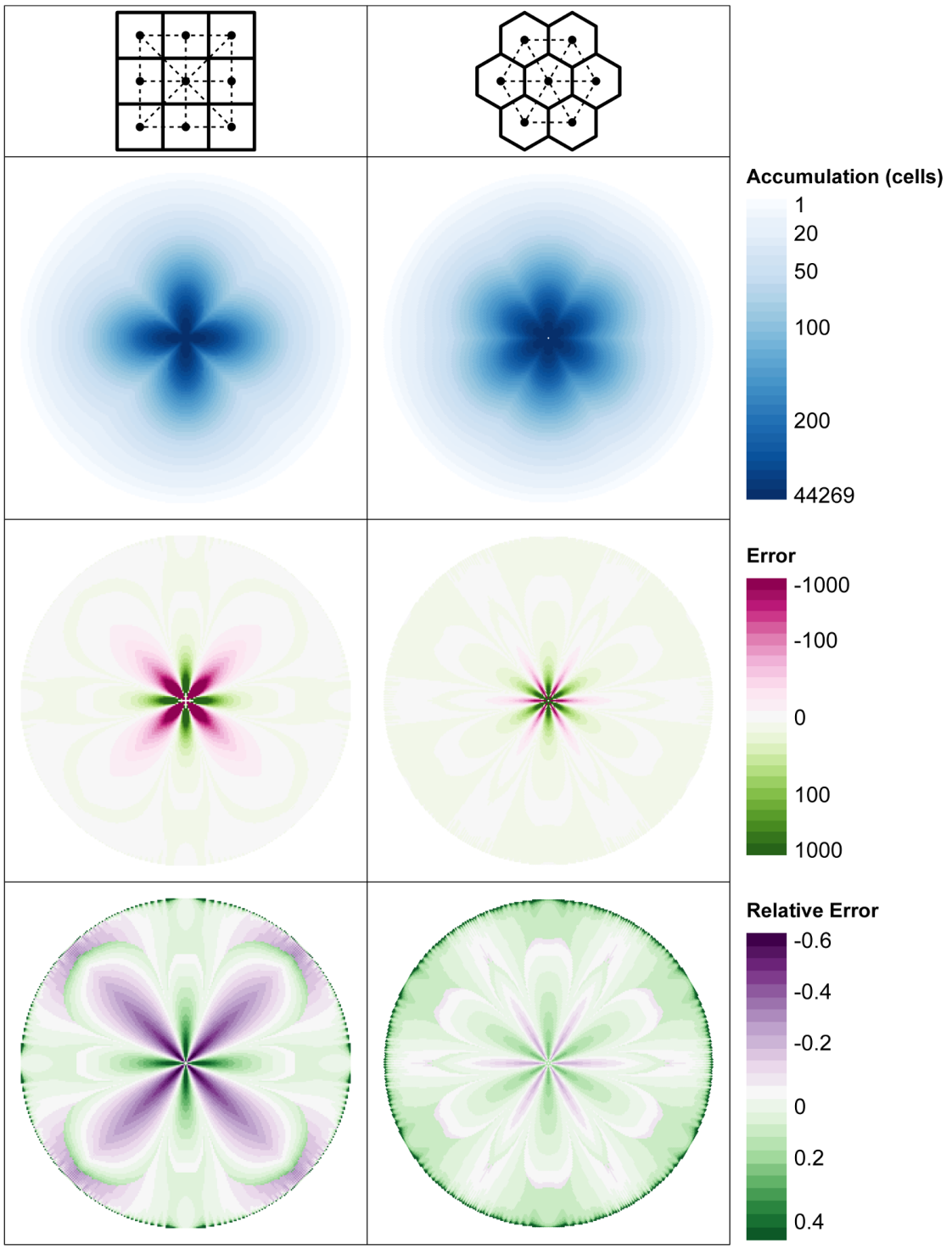


Figure 9.17.: $D\infty$ Angular accumulation on concentrative cones.

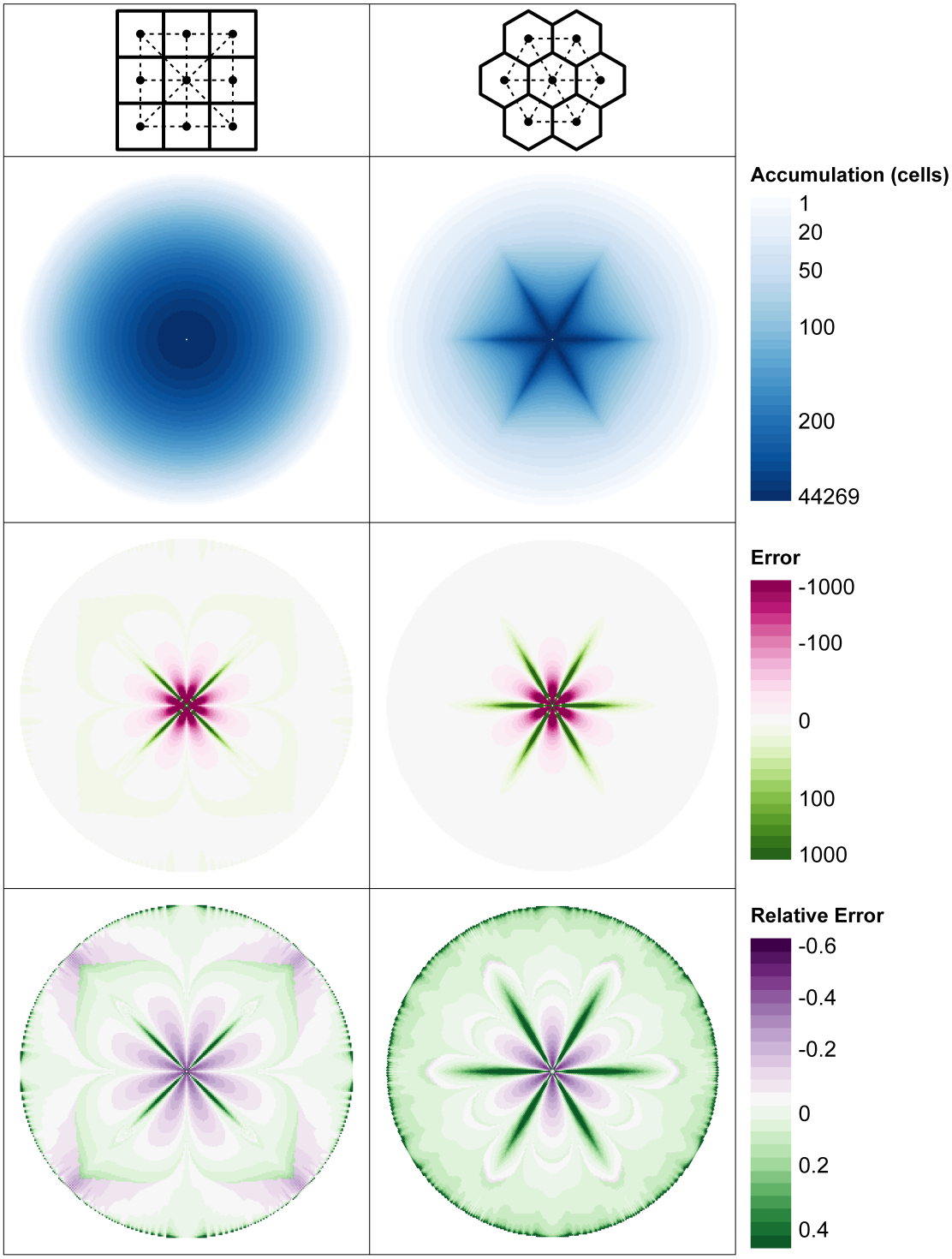


Figure 9.18.: $D\infty$ Areal accumulation on concentrative cones.

D∞ Accumulations Metrics

The trimmed $RMAE$ are summarised in Figure 9.19. Unlike the Dx results, dispersive cone accumulations are not consistently less accurate than concentrative. There is also no clear trend between RHSM-hex and RHSM-rec regarding which algorithm is closest to the analytical result. Angular does well in dispersive cones; however, Linear is more consistent across the geometries and surfaces tested. Areal performs poorly in dispersive cones.

The Relative Errors for each cell are plotted against analytical angle in Figures 9.20, and 9.21 for dispersive and concentrative cones respectively. Dispersive results are generally less reflectionally symmetrical, with the symmetry breaking down for RHSM-hex near the facet boundaries. RHSM-rec datasets are more accurate near the shorter facet edge and less accurate near the longer facet edges. Whereas, RHSM-hex is more accurate at the facet boundaries and less so in the centre of the facet. For comparison, expected relative errors modelled using Equations 9.13 to 9.16 are shown in Figure 9.22.

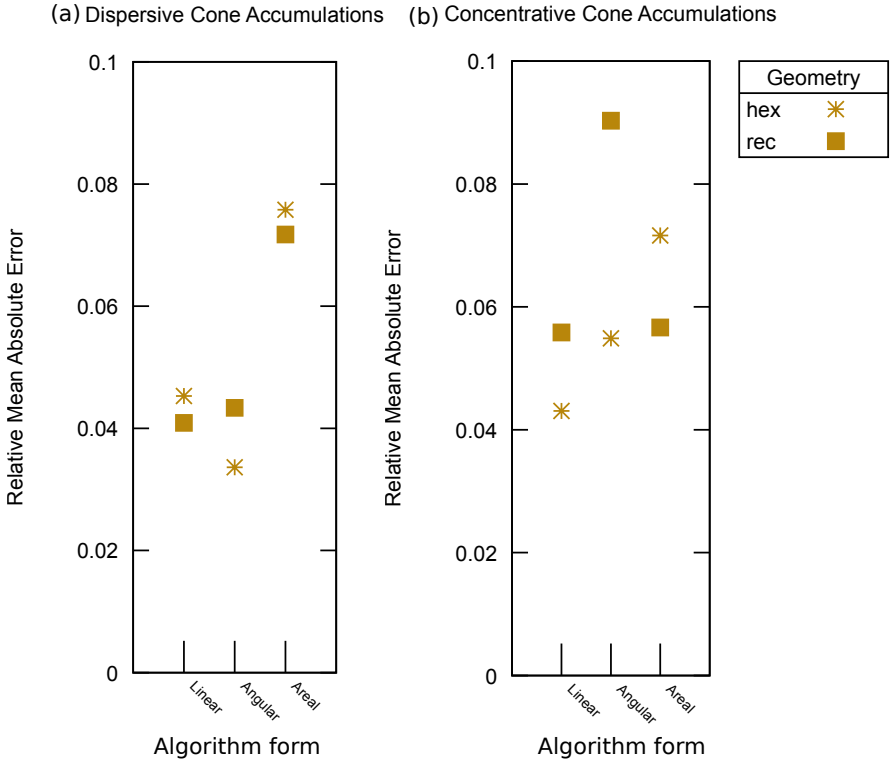


Figure 9.19.: $D\infty$ accumulations trimmed $RMAE$ summary for three versions of $D\infty$ using RHSM-hex and RHSM-rec. The $D\infty$ algorithm is on the x-axis and $RMAE$ on the y-axis. Dispersive cone results are shown on the left and concentrative cones on the right.

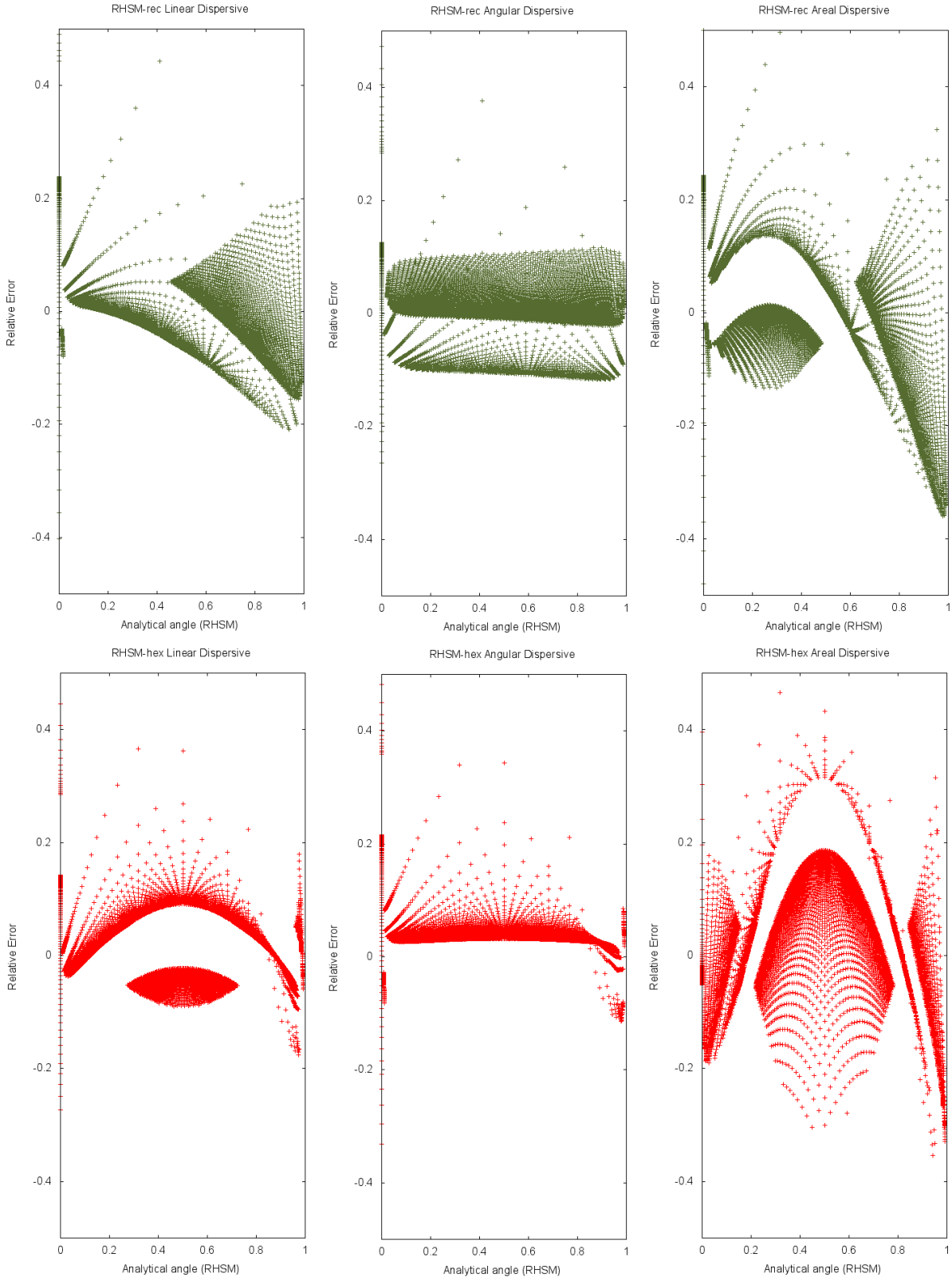


Figure 9.20.: $D\infty$ accumulation Relative Error for each cell in the array for the dispersive cone. The x-axis is analytical angle in RHSM units, the y-axis is the Relative Error. The three forms of $D\infty$ algorithm form the columns: Column 1 is Linear, Column 2 is angular, and Column 3 is areal. The top row in green is RHSM-rec, the bottom row in red is RHSM-hex.

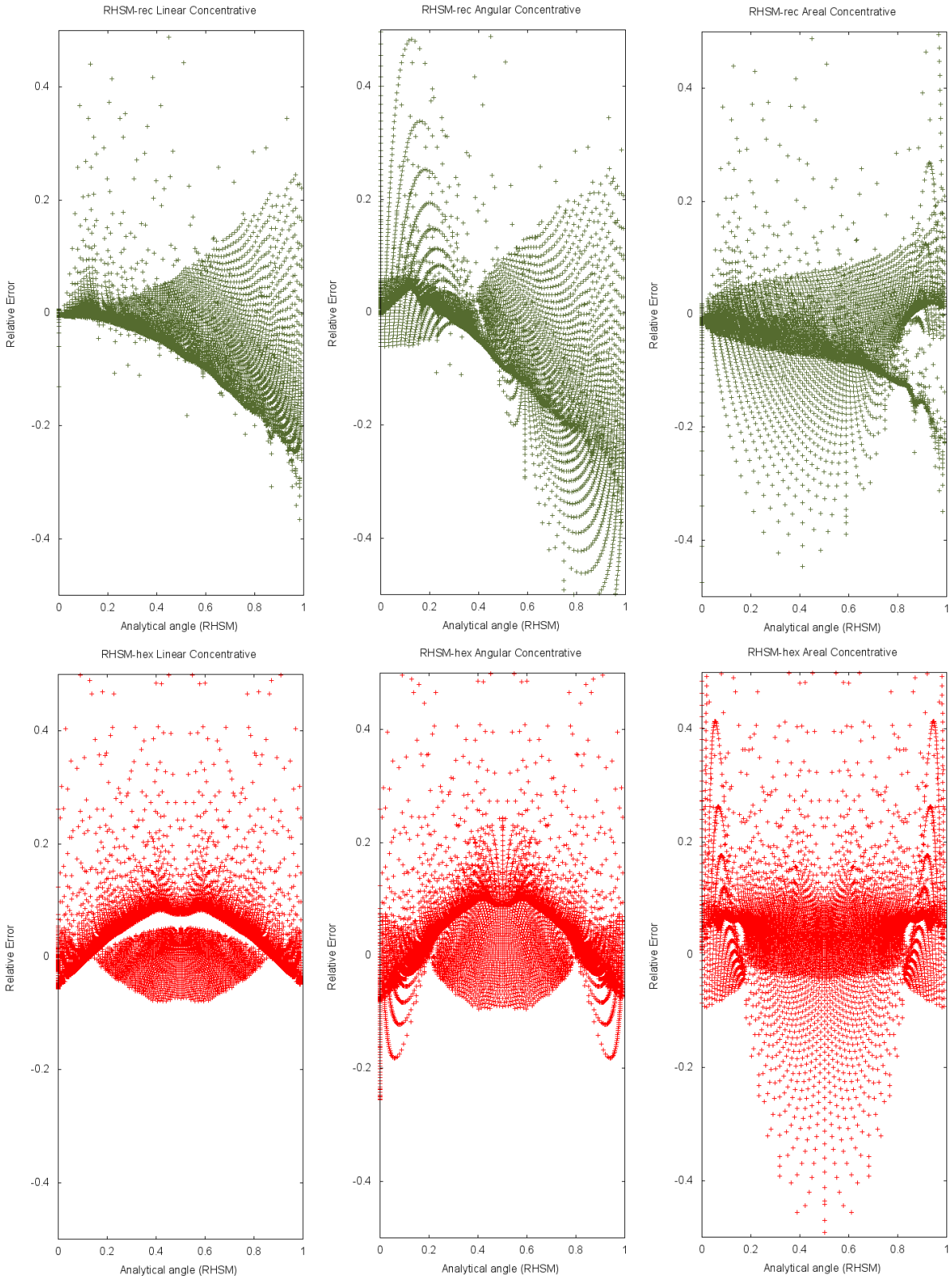


Figure 9.21.: $D\infty$ accumulation Relative Error for each cell in the array for the concentrative cone. The x-axis is analytical angle in RHSM units, the y-axis is the Relative Error. The three forms of $D\infty$ algorithm form the columns: Column 1 is Linear, Column 2 is angular, and Column 3 is areal. The top row in green is RHSM-rec, the bottom row in red is RHSM-hex.

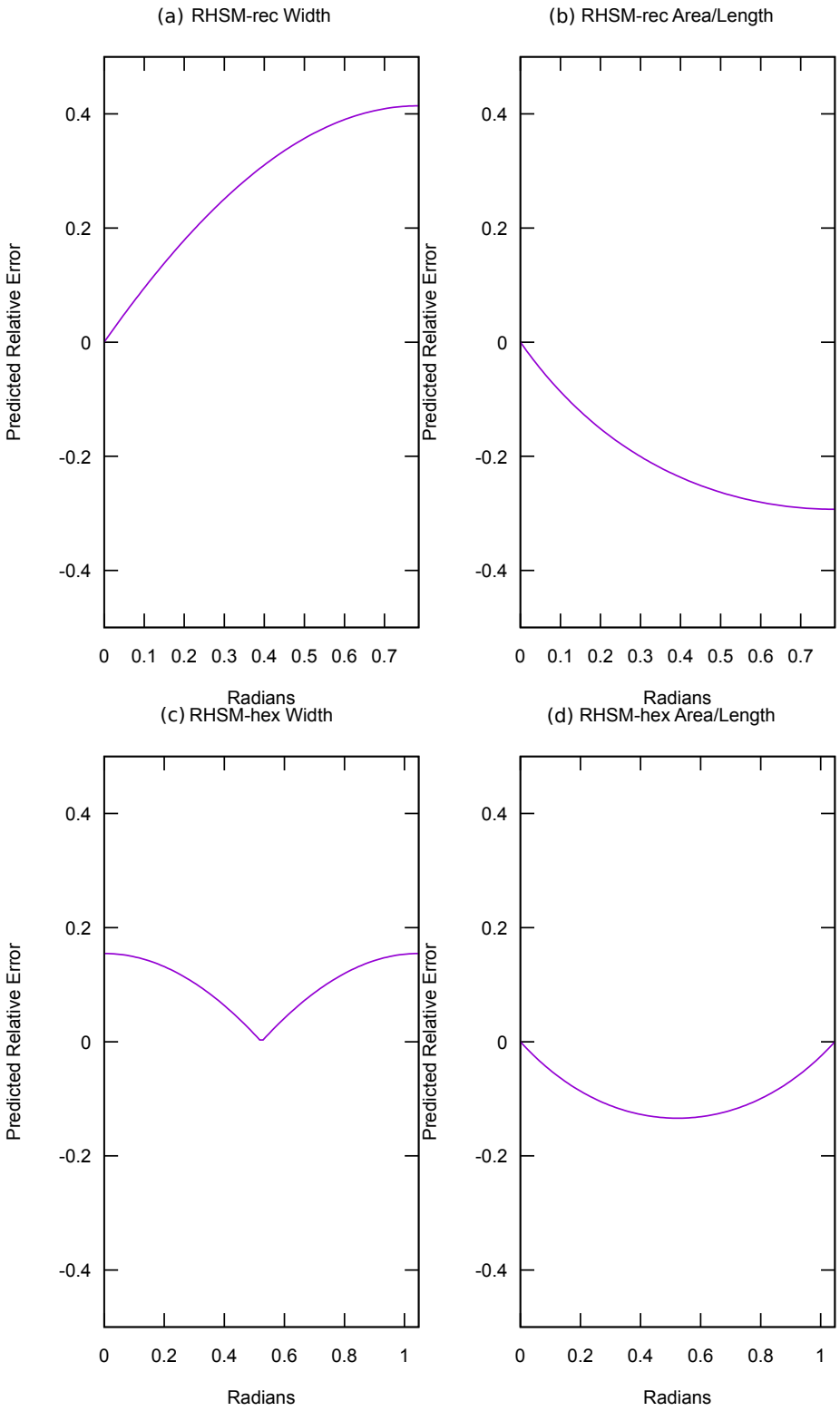


Figure 9.22.: Modelled Relative Errors from cell dimensions. (a) models RHSM-rec option 2 (projected width) from Equation 8.5.1, (b) models RHSM-hex option 2, (c) models RHSM-rec option 4 (Area/Length), (d) models RHSM-hex option 4. Due to the 90 degree rotational symmetry, width and flow length are the same for rectangular cells but not for hexagonal cells.

9.3.4. Discussion

The ideal result of an accumulation on a cone surface would have circular accumulation contours, with no variation in accumulation along the perimeter of any circle centred on the cone centre. The analytical determinations of a in Figures 9.11 and 9.12 are good representations of the ideal accumulation result. $D\infty$ accumulations depart from this isotropic ideal in ways that relate to the geometry of tessellation that are discussed below.

The Relative Errors of concentrative $D\infty$ accumulations in Figures 9.16, 9.17, and 9.18 exhibit edge and centre effects. The edge of the dataset is a poor approximation of a circle due to being composed of regular tessellations. Near the edge of a concentrative cone, the accumulations are small consisting of only a few cells, therefore, there are significant differences between between the RHSM accumulation, which consists of whole cells and the calculation of a, which assumes a circular boundary. Centre effects are exacerbated by the ordered nature of the cone. Errors accumulate, therefore, even very small biases will become large toward the centre of the cone.

The relationship between Specific Catchment Area and flow accumulation

The mid slope section, away from the edge and centre, also does not fit the isotropic ideal. Subsection 8.5.1 proposed that inaccurately modelling the contour length (l) implied by the conversion of (A) into a may cause anistropic behaviour if the estimate of the contour length does not vary with direction. Four options were proposed to model l and graphed in Figure 9.22. Although the general shapes of the errors suggest that there may be a relationship, Options 2, 3, and 4 do not fit the observed errors. Option two shows greatest errors where there should be least, although option 3 modelled errors are not shown, they would be similar to option 2. Option 4 models RHSM-rec errors effectively; however, it does not match RHSM-hex. Unlike the RHSM-rec, RHSM-hex option 4 is not simply a reflection of option 2 across the x-axis because the flow length and flow width are negatively correlated in regular hexagons but identical in rectangles.

Another point that argues against correcting for angle is that, in cone surfaces, all upstream cells have the same angle of incidence or at least are within the same range, but this is not the case in natural landscapes. Therefore, one should not apply a correction to a calculation to account for geometry unless one intends to consider the geometry of all upstream cells, i.e. geometry must be considered during accumulation not after. In reality the geometry effects will probably average out in complex terrain so mean cell width is the most appropriate way of estimating contour length. In fact, the evidence suggests

that the algorithm $D\infty$ works most effectively in situations where the flow direction is aligned with the point set structure. This analysis does not show whether these effects are significant in less structured surfaces.

In summary, there is no justification found here, empirically or theoretically, to consider the variation of flow width (contour length) with direction when converting flow accumulation to Specific Catchment Area in regular tessellations. There are systematic biases in the $D\infty$ flow routing algorithm that are greater than any bias caused by flow width varying due to the direction of flow relative to grid orientation. In other words, errors in the area (A) outweigh the errors in contour length l. Therefore, Option 1: the application of mean cell size, is the most appropriate option.

Observed lack of symmetry

A puzzling result was the lack of symmetry observed most clearly in the RHSM-rec results in Figure 9.14 and also when comparing the symmetrical results for RHSM-hex on concentrative cones in Figure 9.21 with the less symmetric RHSM-rec results on dispersive cones in Figure 9.20. These effects are observed in the anisotropy of errors but are not of sufficient scale to be observed in the $RMAE$.

Seeking an explanation for this phenomenon, a close examination of flow direction arrays revealed that dispersive results are not exactly opposite (π radian different from) the concentrative results, with the greatest discrepancy near to the centre of the cone. The discrepancy between the dispersive and concentrative directions is greatest where the $D\infty$ facet edge aligns with the concentrative direction.

Due to the symmetry of the cone, an angle that is equally subtended from the facet edge but on the opposite side would have the same slope. Critically, this direction is on a different facet so will attribute flow to different neighbours; which neighbours receive flow is governed by how the algorithm breaks the tie. The algorithm was implemented to select the first direction when there is a tie, based on the assumption that this would be very rare. However, on dispersive cones it is not rare, and in fact occurs frequently, inevitably generating the asymmetry observed.

This exposes a failing of the conceptual model. Only a single direction is specified for each cell. However, on dispersive surfaces a range of directions are the steepest downslope. If this were not the case, water would not disperse under the kinematic assumption. The algorithm is randomly picking one of the directions from within the dispersive range and assigning that as the flow direction depending on the order in which the algorithm assessed cells. This causes asymmetry if the dispersion crosses facet boundaries by distributing flow to different neighbours.

Dispersion is a constant bifurcation; on a divergent surface every point is a ridge point with divergent flow paths. The planar assumption inherent in the $D\infty$ triangulation represents this divergence as two possible flow paths on different facets. As the surface gets less divergent further from the centre of the cone, the divergent results narrow and approach the direction which is the opposite of the convergent result or, indeed, the analytical result, which is also affected by this conceptual weakness.

There is a temptation to resolve the divergence asymmetry by averaging the two possible solutions in divergent surfaces but this is not realistic because it effectively assumes that water flows along ridges if the asymmetry is generated by the circumstances described above in divergent terrains. A more conceptual robust solution would be to adapt the accumulation algorithm to correctly model the dispersion possibly by permitting flow to more than two neighbours.

9.4. Dx and $D\infty$ direction compression and accuracy on cone surfaces

9.4.1. Overview

This experiment investigated the reduction in cell numbers (compression) and loss of accuracy (error) associated with variable density LoD realisations for Dx and $D\infty$ flow direction arrays generated on a mathematically defined concentrative cone surfaces. Only RHSM-rec and RHSM-hex were investigated. Dispersive cones were not investigated, however, not withstanding the lack of symmetry identified in Section 9.3, dispersive cones would have the same compression and accuracy. The decision rule applied to determine the realisations set a tolerance value that is the maximum allowable divergence between the value in the sparse realisation LoD and its finest children in the base dataset. In this experiment, the datasets were not limited to a circular shape.

The tolerance value for Dx was 0. Dx compression was plotted against cardinality (the number of cells in the base layer). Increasing cardinality reduces the rate of change between neighbouring cells. The dilution of accuracy on Dx arrays was only determined for RHSM-hex surfaces; there is no dilution of accuracy for RHSM-rec. The accuracy for RHSM-hex changes due to the difference between the directions possible on different LoDs. The results showed that the Dx dilution of precision was significant for the RHSM-hex, which indicates that restricted flow direction algorithms are not effective for RHSM-hex. The results of the Dx analysis were published in Wright et al. (2014).

A range of tolerance values were applied to $D\infty$. For both the RHSM-rec and RHSM-hex, the percentage of compression is highly sensitive to the tolerance value applied in the decision rule. Loss of accuracy was plotted against tolerance. Unlike $D6$, $D\infty$ RHSM-hex flow directions can be upscaled into coarser LoDs without alteration because all directions are permitted.

9.4.2. Methods

Dx Methods

To investigate Dx compression, a total of 14 data sets were created. Seven datasets were created for each of RHSM-hex and RHSM-rec with the number of cells in each dataset equal to $\mathfrak{a}^i \{i \in \mathbb{Z} | 1 \leq i \leq 7\}$, where $\mathfrak{a}$ is the aperture of the RHSM (7 and 9 for RHSM-hex, and RHSM-rec respectively). Each dataset can support a maximum of $i + 1$ LoDs with the coarsest LoD being a single cell. The aggregation value was equal to i. A cell size of 1 was used for all datasets. However, due to the self similarity of the underlying cone, the cell size is immaterial. In contrast to other trials, this experiment conceptually holds the extent constant and makes resolution finer, thus increasing cardinality.

To generate the synthetic elevation values, the HIP ordinate for each cell in the tessellation was converted into (x, y) coordinates, from which the elevation value (z) was determined using Equation 9.1 for concentrative cones. Dx flow directions were calculated for each dataset. Pyramid layers were formed using the spatial average scaling rule and an error pyramid was created to express the maximum divergence from the value of the finest children (Subsection 8.4.2).

Each of the flow direction arrays was then compressed to form a variable resolution surface. The decision rule was to select the coarsest resolution in which the following was true.

$$Ei_{hip} = 0 \tag{9.17}$$

Where Ei_{hip} is the error value at level i at location hip. The compression achieved in each case was determined as a percentage using:

$$\frac{Cb - Cs}{Cb} \times 100 \tag{9.18}$$

where Cb represents the number of cells at the finest level of resolution and Cs represents the number of cells in the compressed dataset.

Vector solutions were determined by calculating the direction from the centre of each cell to the centre of the cone using Equation 9.7. The accuracies of the flow direction arrays were determined for the finest resolution datasets and, for RHSM-hex, the variable resolution results. An accuracy measure for each cell was calculated as the difference between the result and the vector result in radians at the centre of the cell. For the simplified dataset the accuracy result was determined by disaggregating the results back to the base LoD and comparing the direction to the vector result. For the RHSM-hex the rotation between levels causes the result to change. For the RHSM-rec and RHSM-tri the accuracy is the same as the base resolution.

$D\infty$ Methods

λ5RHSM-rec and λ6RHSM-hex flow direction arrays were determined using the $D\infty$ algorithm for a concentrative cone surface (Figure 9.1). Cell sizes were weighted so both the RHSM-rec and RHSM-hex cells had an area of 1m^2. The study area was clipped to a 240m diameter circle. Thirteen variable density realisations were generated for each geometry with tolerances ranging from 0 to $\pi/8$ radians at $\pi/96$ radian intervals.

The compression percentage was determined for all datasets using Equation (9.18). The Relative Mean Absolute Error (RMAE) was determined for each realisation relative to theoretical results and plotted against tolerance.

9.4.3. Results

Dx Results

Figure 9.23 displays the flow direction arrays and the error in radians of a 5λ RHSM-hex on the cone surface and compares it to the base case. The percentage of aggregation achieved for datasets ranging from two to seven LoDs are given in Figure 9.24. The percentage of aggregation increases rapidly with increasing cardinality of the dataset.

Figure 9.25 compares the accuracy of the hexagonal sparse realisations with the base case at the finest level of resolution. These results show that the differences between the simplified datasets and the vector solutions are greater than the differences between the base case and vector solutions across all LoDs as indicated by the mean being further from 0, the standard deviation being greater and the extremes being further from 0. Using the RHSM-rec or RHSM-tri the results will be the same as for the base case, therefore,

they are not shown. The results show a significant deterioration in accuracy when using hexagonal sampling. An ANOVA test at 95% significance shows significant degradation for all but the L3 results, where the sample size is small and little simplification achieved (see Table 9.1).

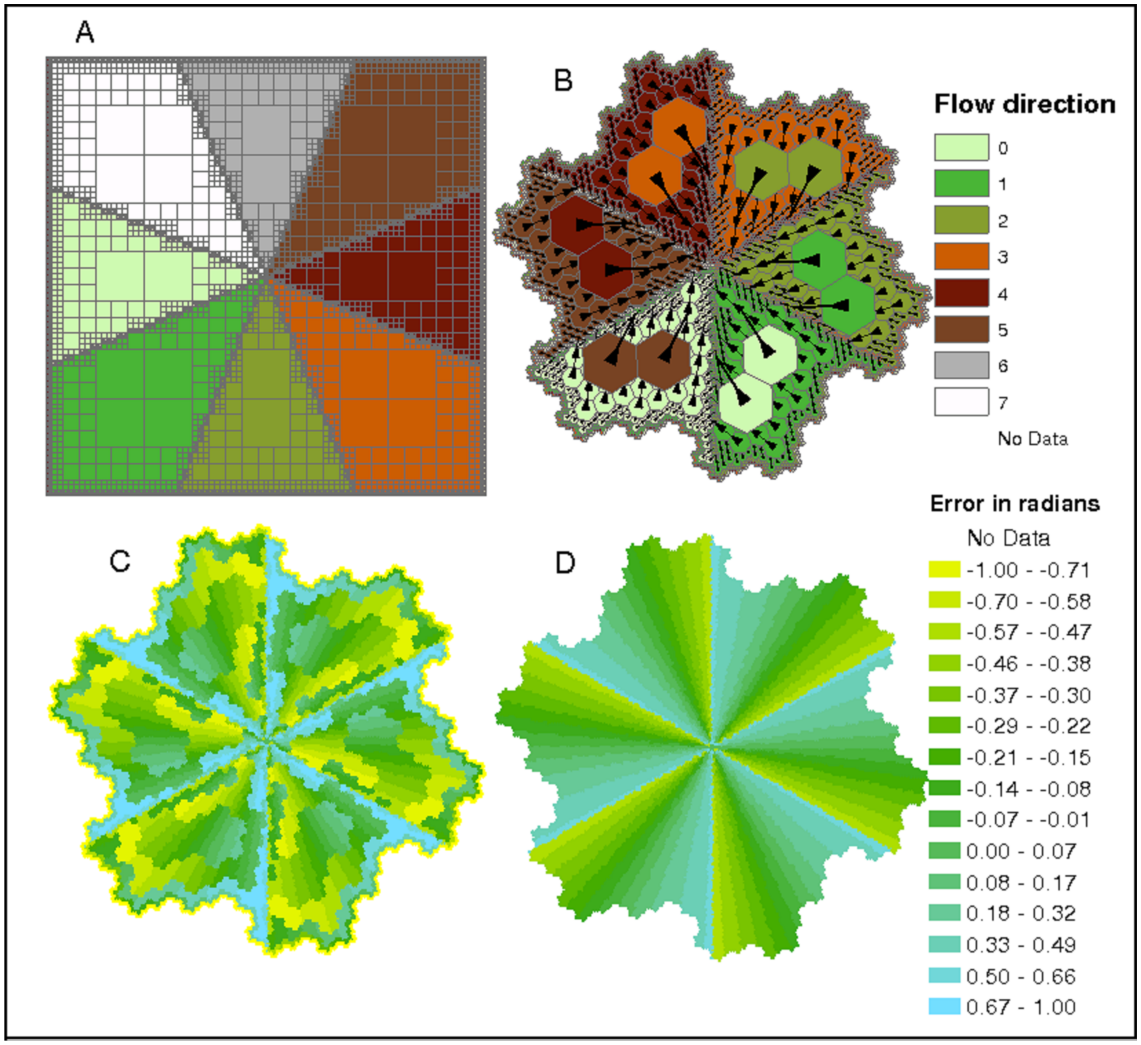


Figure 9.23.: Dx flow directions and errors in radians on a concentrative cone surface. (A) shows the simplified RHSM-rec flow direction. (B) shows the simplified RHSM-hex flow direction. The flow direction values are measured anti-clockwise from the direction of the base vector in multiples of 1/4 and 1/3 radians for the RHSM-rec and RHSM-hex respectively. However, because the direction of the base vector in the RHSM-hex rotates between levels, the directions are also indicated with black arrows. (C) shows the direction errors after forming the RHSM-hex variable resolution flow direction. (D) shows the RHSM-hex direction errors at the finest resolution.

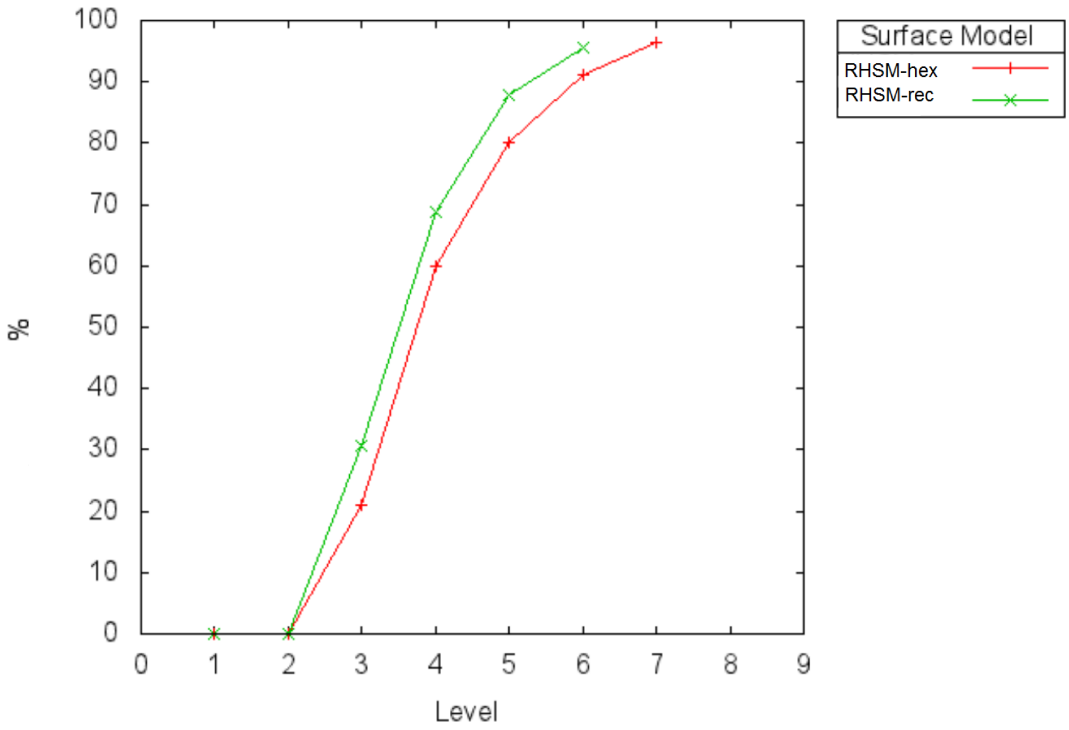


Figure 9.24.: The number of cells removed relative to the number of cells in base resolution versus level for the RHSM-rec and RHSM-hex applied to a Dx flow direction array on a concentrative cone surface.

Table 9.1.: Significance of dilution of precision for Dx RHSM-hex on concentrative the concentrative cone surface. An F statistic greater than 19.5 is significant at the 95% confidence level.

LOD	n	F statistic	95%	Significant
L7	816985	4862.86	19.5	Yes
L6	115465	3165.64	19.5	Yes
L5	16081	283.18	19.5	Yes
L4	2161	21.26	19.5	Yes
L3	265	2.92	19.5	No

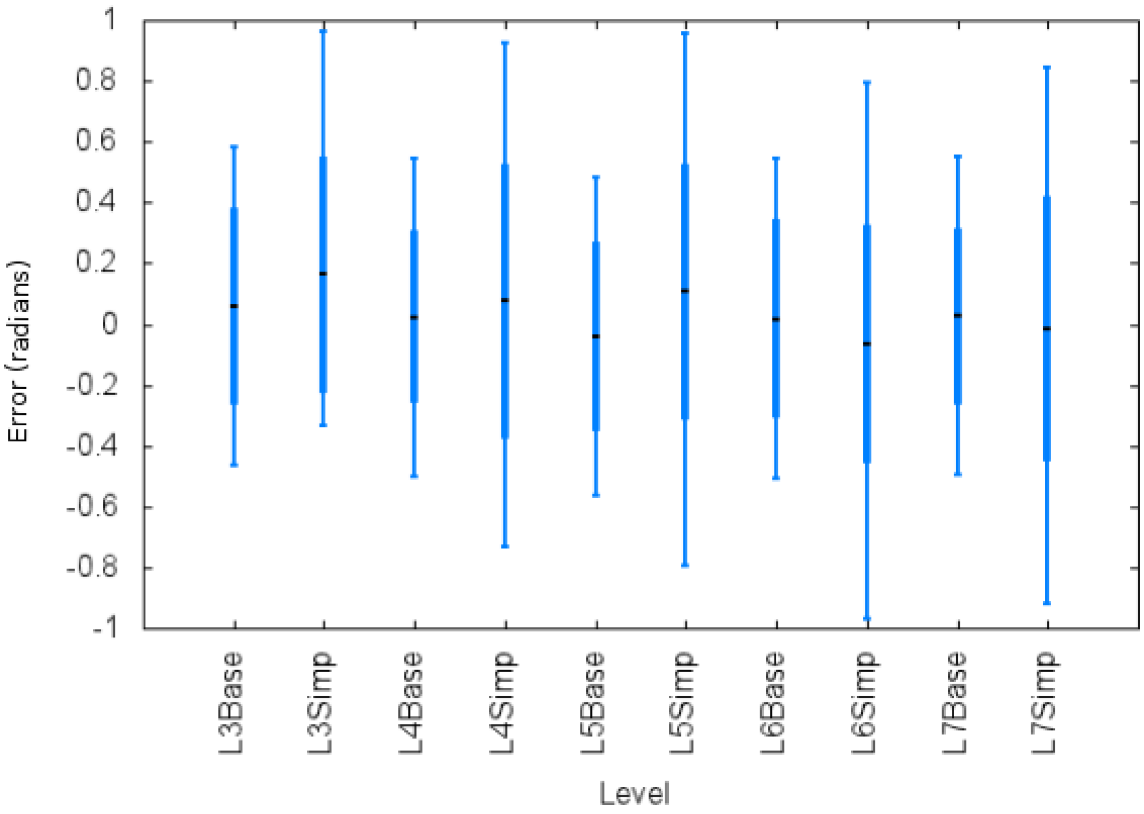


Figure 9.25.: : Box and whisker plot of the dilution of precision for Dx RHSM-hex on concentrative cone surfaces. The mean (black line) +/-1 standard deviation (thick blue line) and extreme values (thin blue line) for the error in radians between the flow direction and the true direction to the centre of the cone. The results for datasets ranging from 3 LoD to 7 LoD are given. LxBase is the result for the full resolution with x LoD. LxSimp is the result after the flow direction simplification on hexagonally sampled data with x LoD.

$D\infty$ Results

Figure 9.26 shows that the orderly structure of the cone surface permitted substantial compression. Error increases as the tolerance increases. However, Figure 9.27 shows that the compression was achieved with comparatively little loss of accuracy. Using a tolerance value of $\pi/24$ in the RHSM-hex generated 94.75% compression with an associated MAE of 0.0336 radians. The respective values for the RHSM-rec were 94.33% and 0.0263. By contrast, the MAE of a flow direction array generated using the $D8$ flow direction algorithm at full resolution was 0.2619 radians for RHSM-hex and 0.1963 radians for RHSM-rec.

Figure 9.27 shows that RHSM-rec has less error for tolerances between $4\pi/96$ and $7\pi/96$ but greater errors above $9\pi/96$. Figure 9.28 displays the flow direction sparse realisations for a tolerance of $6\pi/96$ and the spatial distribution of the errors associated with those sparse realisations. Smaller cells are present in the realisation at the centre of the cone and at the edge.

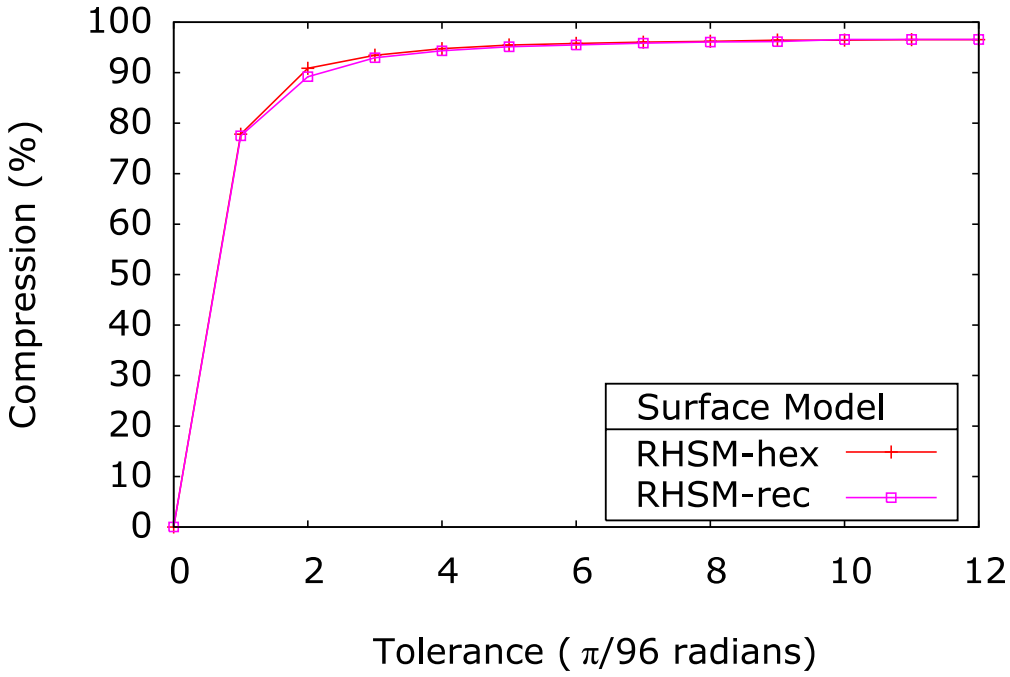


Figure 9.26.: Compression percentage for $D\infty$ concentrative cone

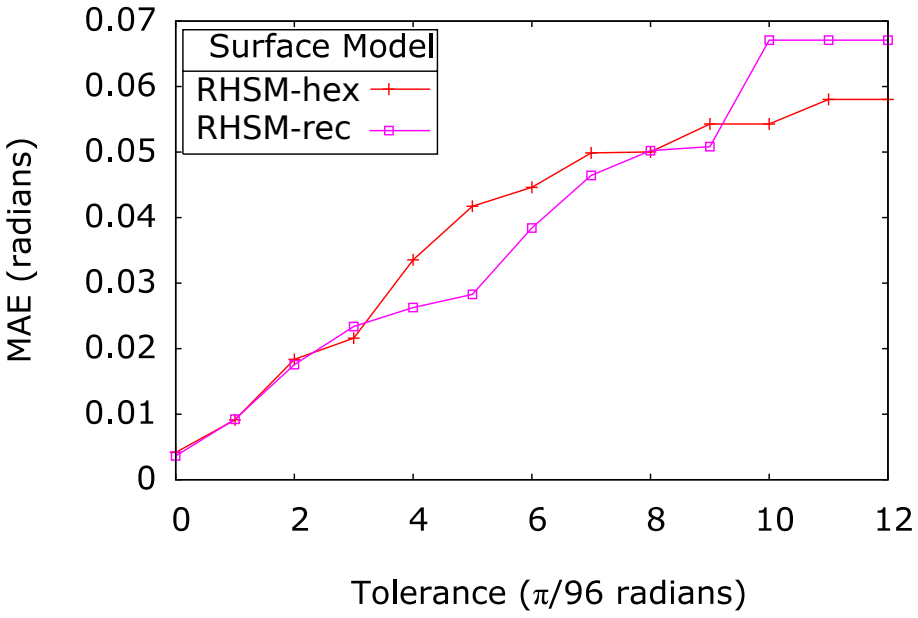


Figure 9.27.: Mean Absolute Error of $D\infty$ flow direction sparse realisations for tolerances ranging from $0^{\pi}/_{96}$ to $12^{\pi}/_{96}$ on a concentrative cone.

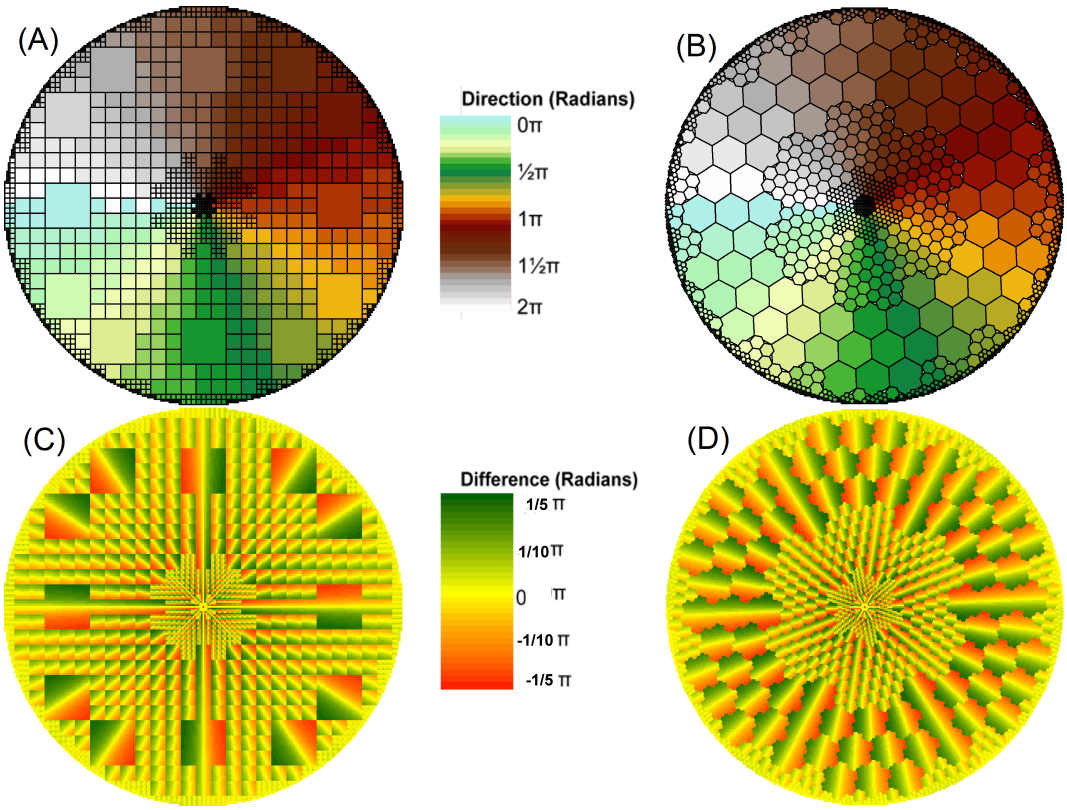


Figure 9.28.: $D\infty$ flow direction and errors for a concentrative cone sparse realisation. (A) and (B) show $D\infty$ flow direction sparse realisations for a tolerance of $6^{\pi}/_{96}$ radians for RHSM-rec and RHSM-hex respectively. (C) and (D) show the spatial distribution of errors for (A) and (B) respectively. The error is shown as the difference from the analytical result in radians.

9.4.4. Discussion

Dx

Compressing the RHSM-hex Dx flow direction array caused significant deterioration of accuracy. The cause of the deterioration is most likely to be the rotation between levels. Adjusting the direction will in some instances improve the result and in others worsen it. However, Table 9.1 indicates that there is a strong bias towards less favourable results. This effect can be explained by the characteristics of the areas that allow simplification when using direction restricted flow direction algorithms.

The $D6$ algorithm divides the cone into six sections, each representing one flow direction. The centre of these sections is the area where the $D6$ direction is the most accurate. The edges of these sections, where the $D6$ changes from one direction to another are where the $D6$ values are the least accurate (see Figure 9.23-C). The decision rule results in coarser levels of detail being implemented where there are large areas of contiguous homogenous cells. Due to the rotation effect, simplifying to a coarser LoD causes the $D6$ direction to change. The contiguous homogenous areas coincide with locations where the $D6$ is most accurate, therefore the simplification is altering the direction where it is more accurate and preserving it where it is least accurate, resulting in an inevitable degradation of results as seen in Figure 9.25. The rotation between LoDs makes restricted flow direction algorithms impractical when using the RHSM-hex.

One simple way to avoid the rotation effect would be to only use the rectangular sampling method of RHSM-rec. However, the other benefits of hexagonal sampling for hydrological analysis would then be lost. The negative results associated with RHSM-hex compression could be avoided by only simplifying the surface when the coarser resolution result is similar to the vector result. This is not generally possible because the vector result is not typically known except when using mathematically defined surfaces. A further possibility would be to aggregate in areas where there is agreement between the fine level directions and a coarser level result defined from separate flow direction arrays calculated on coarser realisations of the surface.

D∞

The unrestricted flow direction algorithm $D\infty$ Tarboton (1997) avoided the problems associated with inter-LoD rotation of RHSM-hex. The tolerance values where RHSM-hex is less accurate (from $4\pi/96$ to $7\pi/96$ in Figure 9.27) correlate with datasets where there is more compression of the RHSM-hex surfaces due to the more efficient packing

of hexagonal tessellations. The increased compression results in larger errors. Toward the higher end of the tolerance range, both datasets are approaching the limit of the number of large cells that can be accommodated as evidenced by the flattening of the lines in Figure 9.27. The largest hexagons that can be accommodated are smaller than the largest squares. However, this is not due to packing efficiency; it is due to the difference in aperture between geometries creating different pyramid scale ranges. For the size of dataset tested, the RHSM-rec happens to have a larger pyramid cell size that can fit within the D8 segments than RHSM-hex.

The presence of smaller cells in the centre of the cone is explained by the increased rate of change of the flow direction. However, the small cells around the edge of the study area, where rate of change is least, are created by the interaction between the RHSM geometry and the circular limit; The edge effects observed in the small cells around the edge are an indication of what would likely occur at suden changes of surface properties such as the edges of built structures. Smaller cells are required to fill in the gaps around the larger cells without extending beyond the edge or discontinuity. This effect is form of the Modifiable Areal Unit Problem (MAUP), that affects vector analysis. Filling edges of polygons with RHSM tiles may offer a structured method to calculate values for modified areal units without relying on ad-hoc polygon slivers. However, this potential has not been explored here.

9.5. D∞ accumulation accuracy by tolerance

9.5.1. Overview

The goal of this experiment was to compare the accuracy of $D\infty$ sparse flow accumulation arrays to mathematically defined base resolution results, in order to asses the loss of accuracy associated with compression. The accuracy was determined for cone surfaces, concentrative and dispersive. The accuracy was determined for the base resolution and for a series of variable density sparse realisations generated by applying the flow accumulation algorithm to the sparse flow direction arrays generated in Section 9.4. The results show that the variable density flow accumulation algorithm is distributing flows unevenly.

9.5.2. Methods

Dispersive and concentrative cone surfaces were defined using Equation 9.1. Flow direction was calculated for each dataset using $D\infty$ and sparse datasets were generated from

the flow direction arrays using tolerances ranging from $0^{\pi}/_{96}$ to $12^{\pi}/_{96}$, i.e. the sparse flow direction arrays from Section 9.4 were used as the input arrays for this experiment. Flow accumulations were determined using the leading coarse accumulation model (Subsection 8.5.3) and the linear apportionment form of $D\infty$.

In order to compare the coarse accumulation with the fine Specific Catchment Area (a), the sparse accumulations were "collapsed" back to the resolution of $L0$ by assigning each cell the value of its sparse parent divided by the difference in cell size (the distance between line adjacent neighbours) between the sparse realisation and base LoD using:

$$A' = \frac{A}{\sqrt{\mathfrak{a}}^{\lambda-\tau}} \tag{9.19}$$

Where A' is the collapsed $L0$ value, A is the sparse value, λ and $\mathfrak{a}$ are the level and aperture of the RHSM respectively and τ is the LoD of the sparse cell.

The errors were determined relative to analytical results using Equation 9.9.

9.5.3. Results

The $RMAE$ errors are plotted against tolerance in Figure 9.29. The accumulation and spatial distribution of the errors for the concentrative and dispersive cones are shown in Figures 9.30 and 9.31 respectively. RHSM-rec concentrative errors decrease as the tolerance increases from $2^{\pi}/_{96}$ to $5^{\pi}/_{96}$. Other combinations of tessellation geometry and cone shape also feature tolerance ranges where errors decrease with coarser resolution.

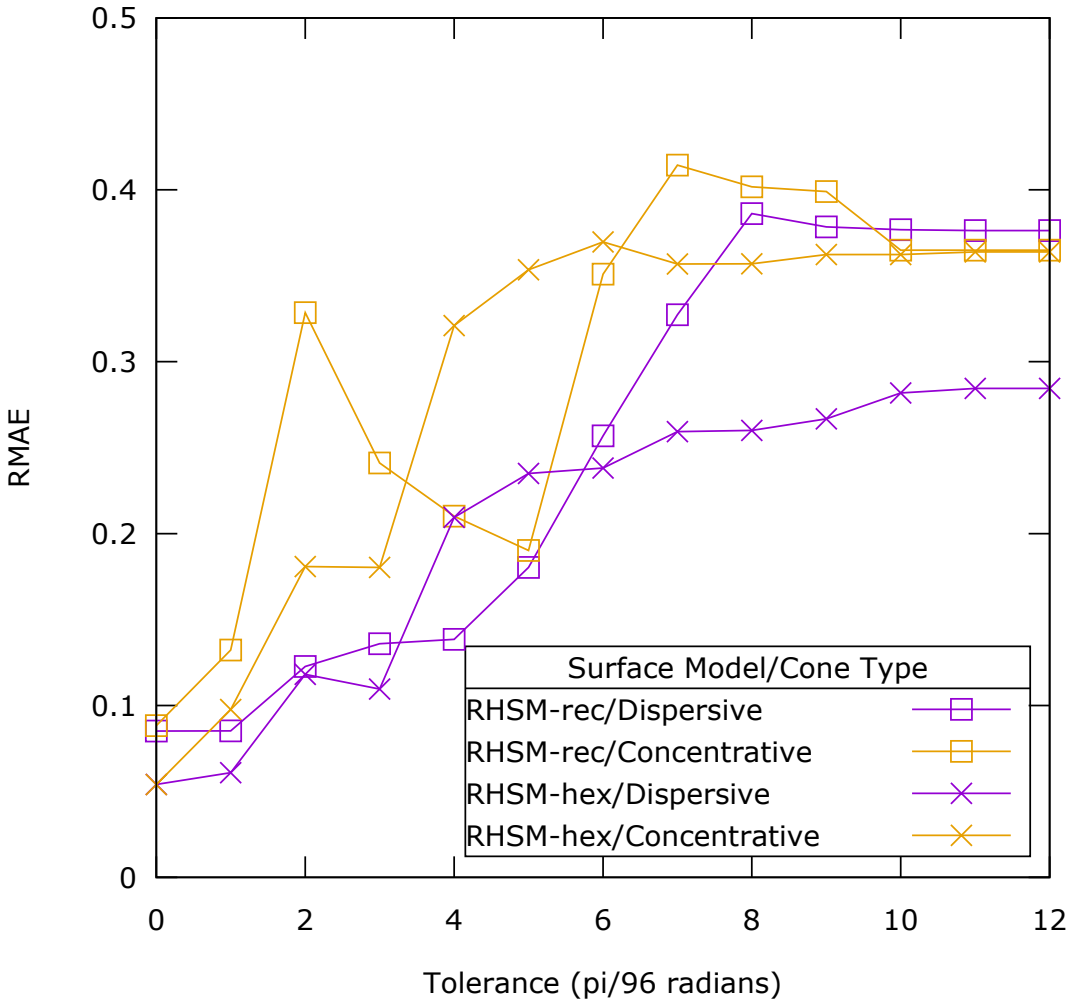


Figure 9.29.: : Relative Mean Absolute Error of $D\infty$ accumulations by tolerance on concentrative and dispersive cones.

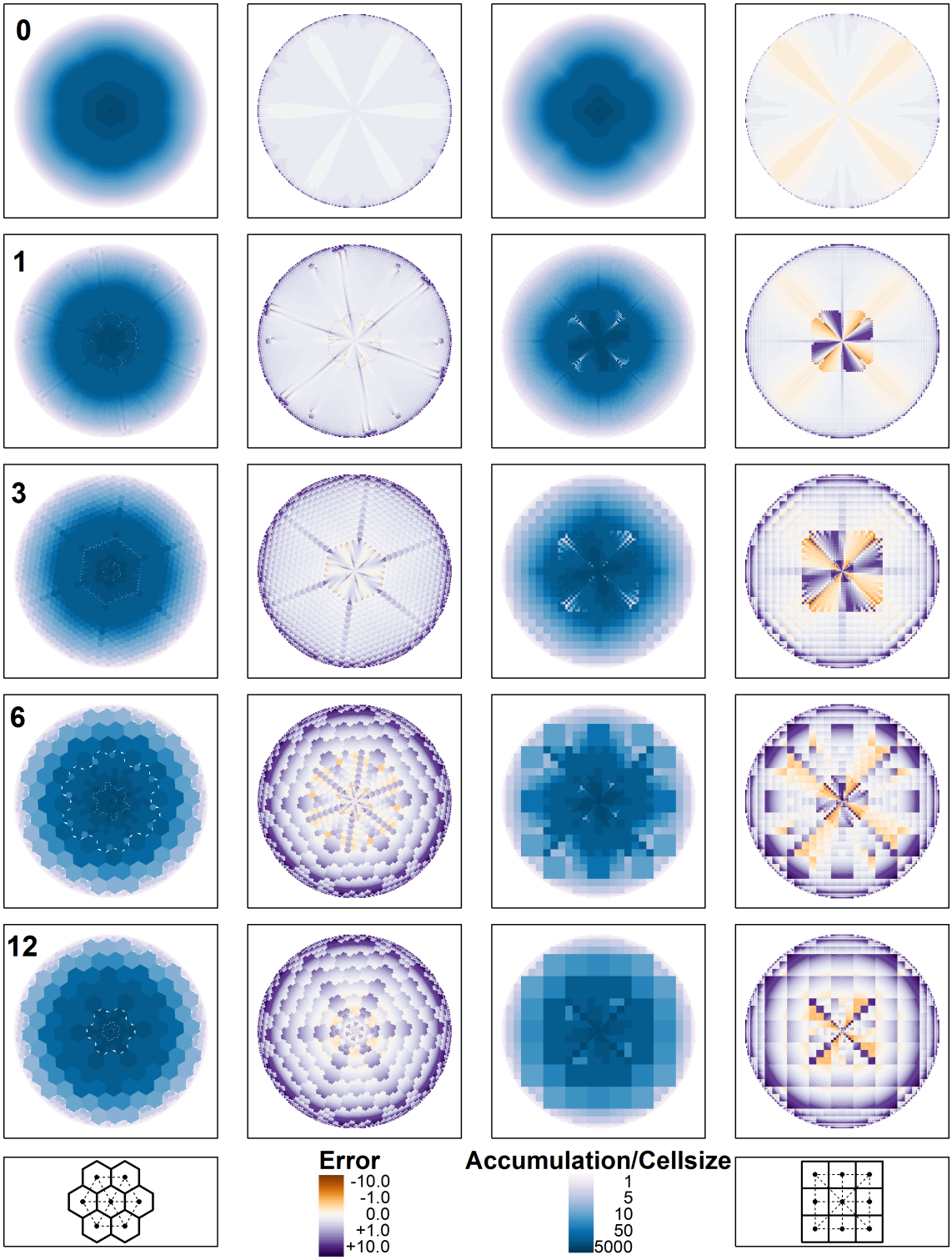


Figure 9.30.: Sparse $D\infty$ accumulations on concentrative cones. The first and second columns are RHSM-hex, third and fourth are RHSM-rec. The first and third columns show accumulation in blue. The second and fourth columns show Relative Error in diverging orange and purple. The numbers in the first column indicate the tolerance (in units of $\pi/96$ radians) for each row. Only selected tolerance values are shown.

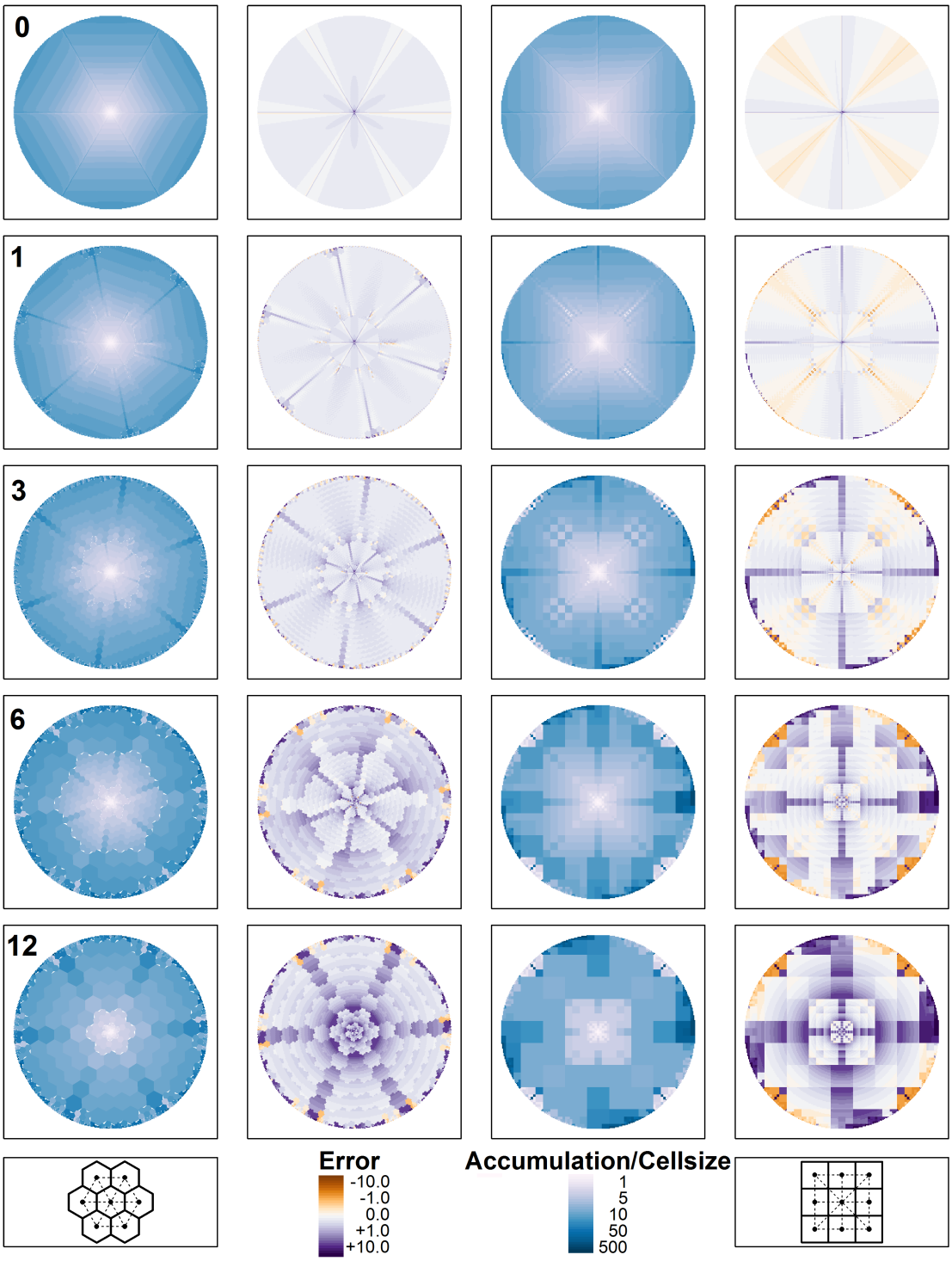


Figure 9.31.: Sparse $D\infty$ accumulations on dispersive cones. The first and second columns are RHSM-hex, third and fourth are RHSM-rec. The first and third columns show accumulation in blue. The second and fourth columns show Relative Error in diverging orange and purple. The numbers in the first column indicate the tolerance $\pi/96$ radians) for each row. Only selected tolerance values are shown.

9.5.4. Discussion

The results in Subsection 9.5.3 indicate that the flow accumulation algorithm is not functioning as anticipated in variable density realisations. Figure 9.29 shows that for RHSM-rec on concentrative cones and to a lesser extent using the other geometry/cone combinations, accuracy improved with increasing tolerance for some intervals of the tolerance range investigated. Increasing the tolerance should reduce accuracy because the larger cells are disguising sub pixel variability The improvement is related to increases in tolerance that reduced the number of changes in LoD along the flow paths; sometimes greater tolerance reduces the number of such changes causing the results to improve, which indicates that the "leading course" method of distributing flows between different LoDs is causing substantial deterioration of accuracy.

Where flow is from coarse to fine some cells receive flow from two neighbours and some from only one; this was not accounted for, resulting in uneven downstream flows. The problems identified with the LoD transitions require addressing through refinement and adaption of the apportionment model described in Subsection 8.5.3.

A second observation indicates a different problem. All datasets have a high mean, which casts doubt on the model used to collapse the results from sparse to fine resolution for assessment. It is important to use an adjustment based on linear cell size, an areal adjustment will reduce the accumulations too much and create large shortfalls. Some overstatement is expected because the accumulation for each cell is added before calculation of accumulation, therefore, the upstream part of coarse cells will include some flow that in a finer grid will be downstream but it is unclear whether the magnitude of error observed (about 30%) can be explained by this effect.

9.6. RHSM $D\infty$ Urban surface test

9.6.1. Overview

This analysis evaluates the relationship between tolerance and compression for RHSM flow direction arrays on three different types of urban surfaces. The three areas selected for the case studies feature different levels of urban development and topographies: a fully urbanised area, a medium density flat suburban area, and a hilly low density suburban area. The greatest compression occurred in the low density hill area where there is a greater percentage of non-building surface and distinct flow directions caused by the hill slopes. Given the poor results for flow accumulation on variable density realisations of cone surfaces, flow accumulations were not determined. In addition, flow accumulation on real surfaces would require an effective pit resolution method. Possible approaches to hydrological conditioning in RHSM are discussed in Subsection 10.3.4.

9.6.2. Methods

All three areas analysed are within the Dunedin urban area and are covered by the Dunedin City Council's LiDAR data and GIS database. Dunedin is a small city of approximately 120,000 people on the shore of an estuary that is typically characterised as a harbour. Dunedin extends from the harbour into the surrounding hills. The three urban areas analysed are named and characterised below, and mapped in Figure 9.32

1. City: A fully urbanised area that includes the Octagon (the central block of Dunedin's Central Business District), extends across flat commercial and industrially zoned land toward the harbour, and includes some of the medium density City Rise suburb on the surrounding hills.
2. Flat: A medium density suburban area south of the CBD that is very flat, close to sea level, features a grid street layout hosting a range of single and multi-family homes typically of one or two stories, with a lot of secondary buildings in yard spaces.
3. Hill: A hilly area in the low density suburb of Wakari, consisting of winding streets and mainly single family detached housing.

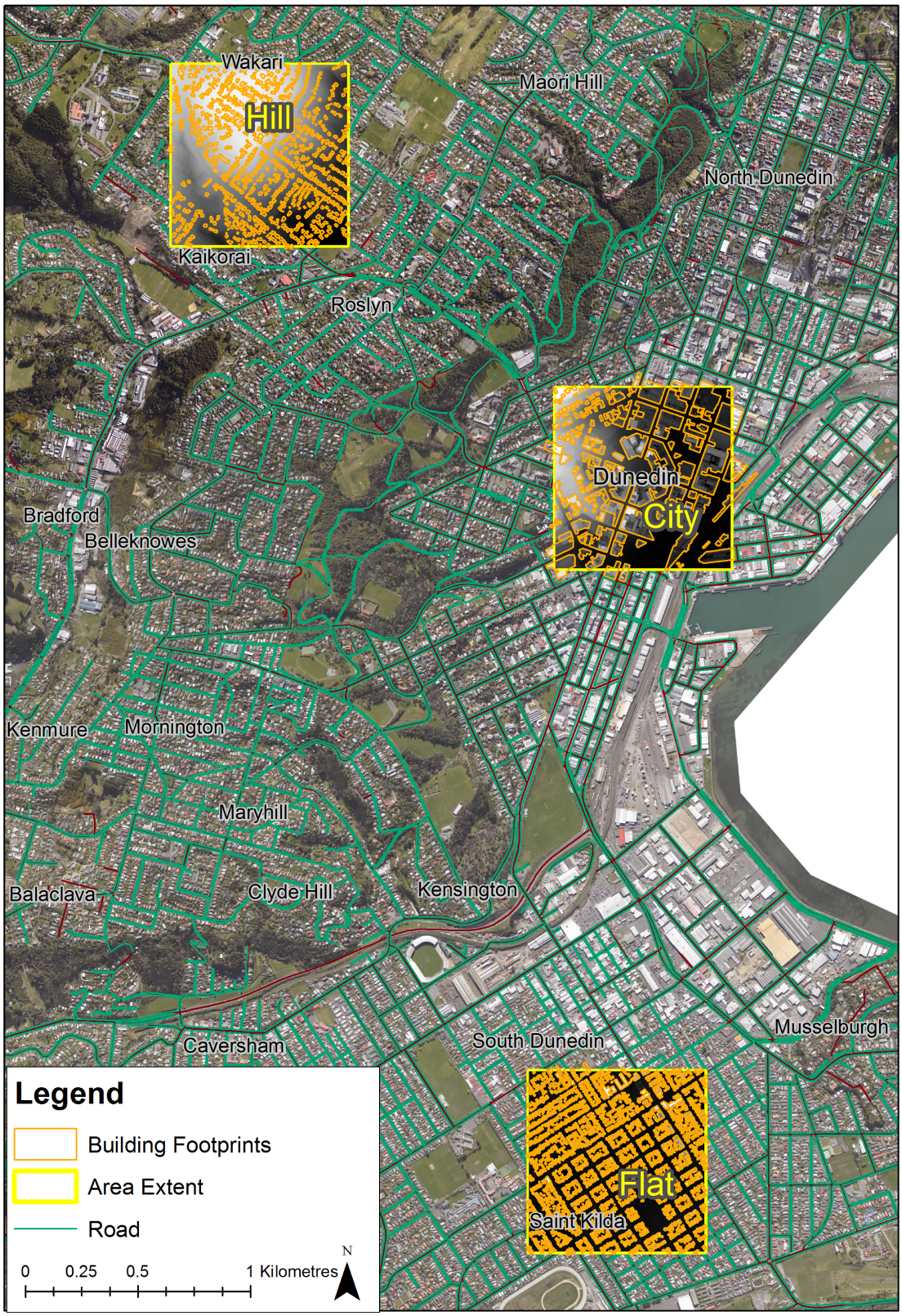


Figure 9.32.: Urban case study areas with DSM shown in the background of the case study areas.

Urban surface flows typically constitute a greater proportion of total run-off than in undeveloped areas. Subsurface flows, although important for ecological systems, slope stability and soil strength, do not generally play a large role in the design of stormwater infrastructure (Wright and Leonard, 2012). Therefore, it is surface flow which holds greatest potential for costs and controls infrastructure design. For these reasons, the subsurface component of urban hydrology is not dealt with explicitly here.

The surface models were created by combining LiDAR generated 1m cell size DTM and DSM rasters. Values from the DSM were used in areas identified as building by visual interpretation of orthophotos and DTM values elsewhere. The resulting surface is predominantly ground and buildings. However, some surfaces that are neither ground nor building will remain where, for instance, vegetation overhangs buildings. The mixed raster surface was converted to a 1m cell size RHSM-rec using the raster values at the centre of each RHSM cell. The RHSM-rec was converted to an equal density RHSM-hex using the Condat method (Condat et al., 2008). The extents of the case study areas were defined by a 6 level RHSM-rec with aggregation value 6 ($6\lambda^9 6\alpha$), which contained 531 441 cells and covered an area of 62.8 hectares. The RHSM-hex dataset was level 8 ($8\lambda^7 8\alpha$). Values in the RHSM-hex that were outside the extent of the RHSM-rec were set to NoData.

Flow direction arrays were determined on the RHSM-hex and RHSM-rec elevation surfaces using $D\infty$. Thirteen variable density realisations were generated for each of the three sample urban surfaces with tolerances from 0 to $\pi/2$ radians at an interval of $\pi/24$ radians. A percentage of compression was determined for each of the sparse realisations using Equation 9.18.

9.6.3. Results

The percentage of compression is plotted against tolerance in Figure 9.33. Flow directions arrays for part of each of the three study areas are shown in Figure 9.34. Compression is greatest in the hill suburb, and least in the flat suburb. Overall rates of compression are much lower than in the structured cone surfaces. For larger tolerance values RHSM-hex consistently displays lower compression.

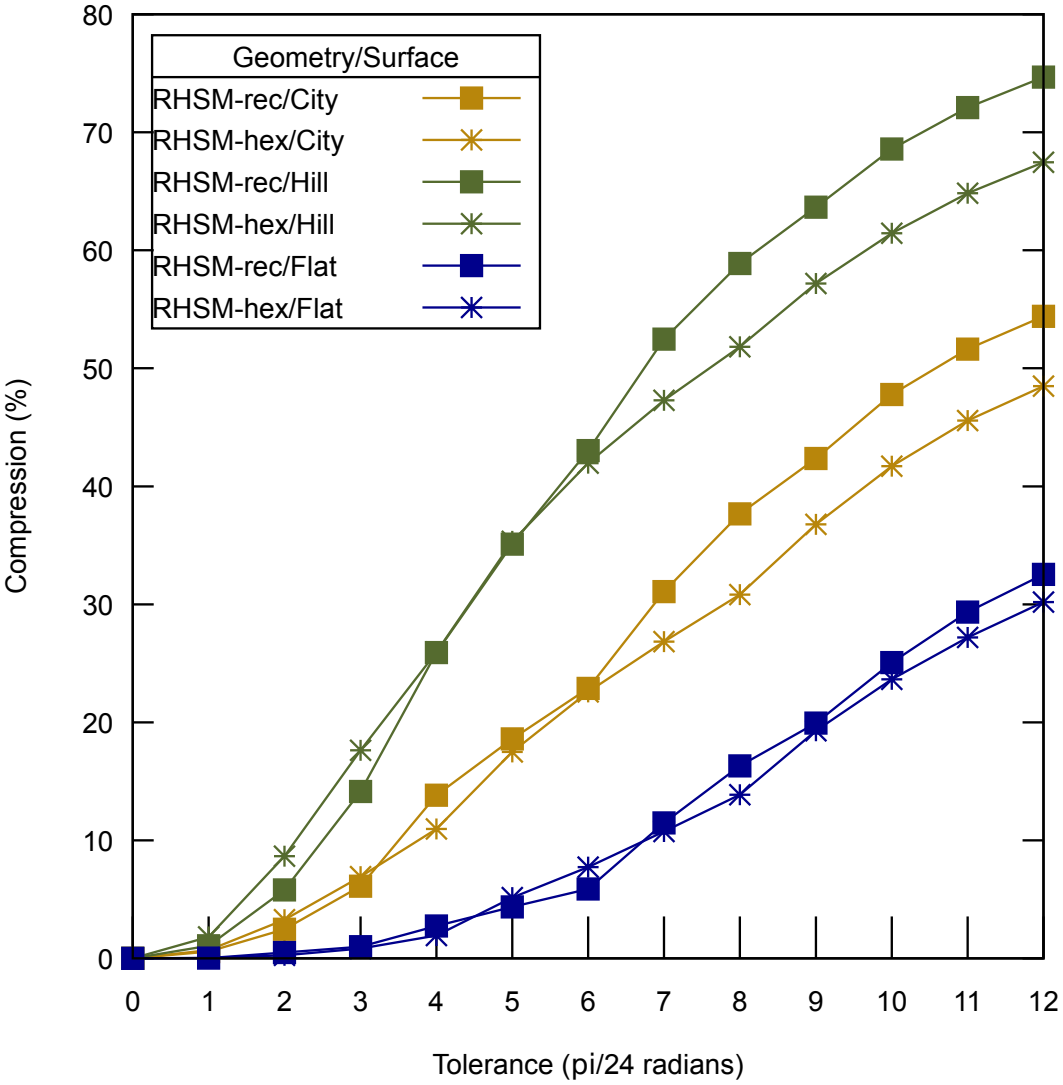


Figure 9.33.: Compression of urban flow direction arrays. The city area is shown in yellow, the hill area green and the flat area blue.

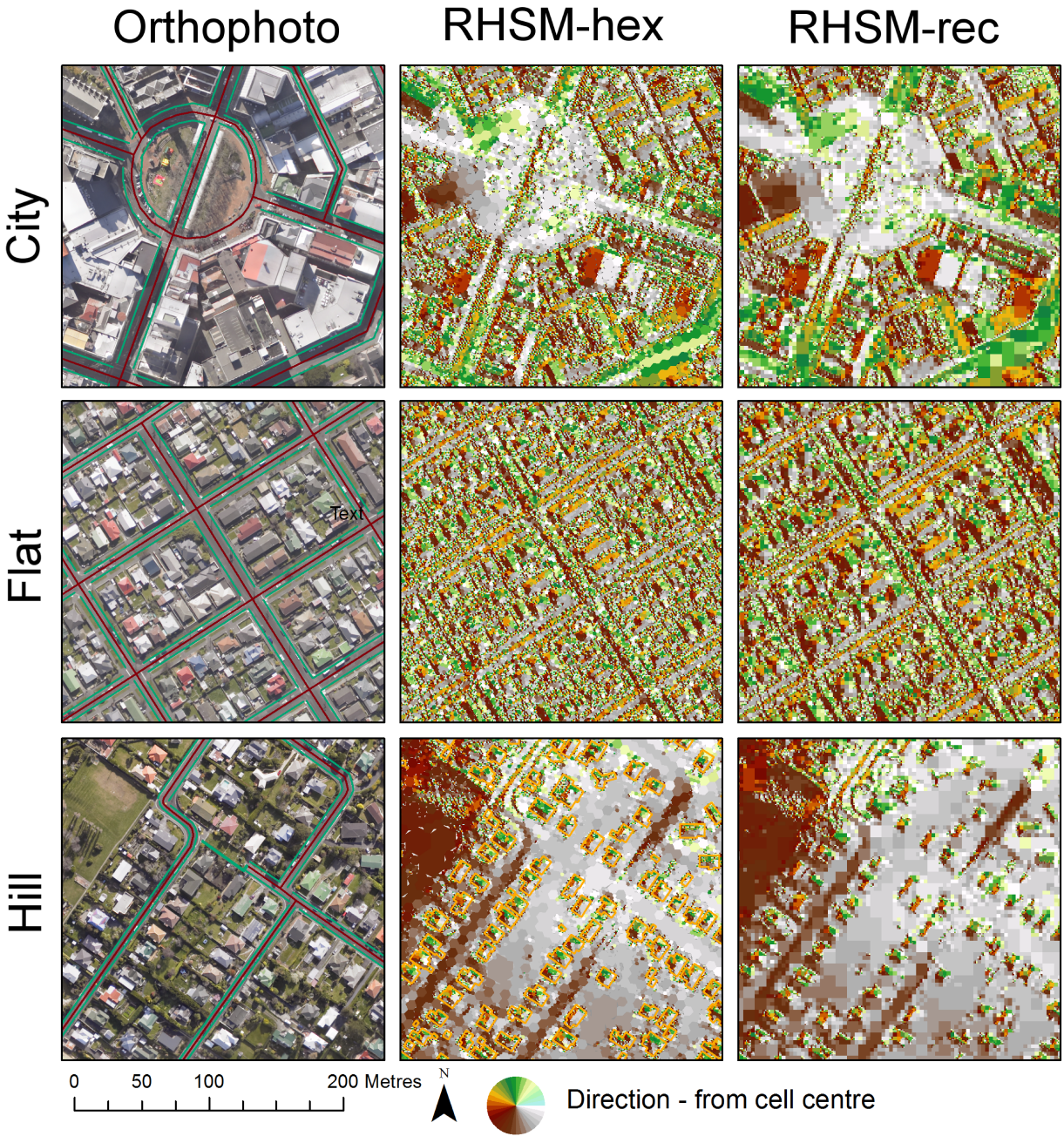


Figure 9.34.: Urban flow directions. Each row of the diagram shows the orthophoto, and RHSM-hex and RHSM-rec flow direction arrays for one of the three study areas. Only part of the study area is shown so that the cell sizes are visible. The flow direction arrays shown are those with the coarsest tolerance value used ($1/2\pi$ radians).

9.6.4. Discussion

Comparing the compression achieved on structured cone surfaces (Figure 9.26) with the compression achieved on unstructured urban surfaces (Figure 9.33) shows that, for given tolerance values, far less compression was achieved on the unstructured surfaces. This trial has demonstrated that flow direction realisations can be simplified using RHSM LOD modelling whilst preserving hydrological features. However, it is not clear whether the compression achieved on unstructured urban surfaces justifies the potential loss of accuracy. Compression could be optimised by undertaking hydrological conditioning before or during LOD modelling. Hydrological conditioning is discussed in a separate subsection below.

The order of compression for the case study surfaces from greatest compression to least was Hill, City, Flat; this is consistent with the following observations.

1. Flat areas do not compress because in flat areas directions are very varied and there are many sinks. Both the varied directions and sinks are plausibly the result of measurement or interpolation noise. Even if the surface variation in flat areas is genuine it may not be necessary to preserve it for flood modelling because surface water will accumulate and fill these small depressions during rain events and, therefore, the kinematic effect of small ground variations on the direction of water flow will be limited.
2. LiDAR derived DTM ground surfaces are smoothed already by the ground classification algorithms, which has the effect of removing noise and allowing more compression. The greatest proportion of ground surface was in the Hill surface.
3. The presence of rapid elevation changes around building roof edges means that only large building roofs are simplified. At finer base resolutions, rooftops, which are generally smooth and have consistent direction could see greater compression. However if resolution became finer still, the texture of, for instance, roof corrugation would begin to prevent compression. This is an example of the importance of considering model intention when defining decision rules. Is the direction of corrugations important to hydrological modelling? If not, the decision rule should be constructed to reflect this.

RHSM-hex surfaces exhibited less compression. Given that this was not seen in the synthetic cone surfaces, the relative decrease in compression for RHSM-hex was probably due to noise introduced by interpolation from RHSM-rec.

9.7. Evaluation summary

This evaluation has assessed the RHSM for the task of hydrological geomorphometry. Hydrological geomorphometry is, by no means, the only possible function of the RHSM. However, it is an example of how the RHSM can identify appropriate scale for modelling and improve understanding of geometric and scale effects in algorithms. The importance of geometry and scale is explicitly recognised by the structure of the RHSM.

Important findings from the individual experiments are catalogued below.

1. Section 9.2 evaluated the behaviour of flow direction algorithms within the three regular geometries. The flexibility of unrestricted directions permitted greater accuracy. However, $D\infty$ directions still had geometric bias artefacts that were related to the geometry of the elevation nodes used by the algorithm to determine direction.
2. Section 9.3 evaluated the behaviour of flow accumulation algorithms within the three regular geometries and explored the relationship between flow accumulation and Specific Catchment Area (a). No benefit was found in considering the perpendicular cell cross section of the flow direction when converting flow accumulation to a. In addition, a lack of symmetry on dispersive cones revealed a conceptual weakness of $D\infty$ flow apportionment in dispersive environments. The various $D\infty$ apportionment algorithms were found to vary in effectiveness between geometries. Although no algorithm performed best in all geometries, linear apportionment gave the best overall performance.
3. Section 9.4 evaluated the effect of scale on flow direction by determining sparse realisations of flow direction arrays. Dx does not compress well in hexagonal tessellations but $D\infty$ worked well in all geometries, which emphasises the value of unrestricted flow directions. The hexagonal grid demonstrated superior compression but also inferior accuracy across a range of tolerance values.
4. Section 9.5 evaluated the accuracy of flow accumulation arrays generated from the variable density direction arrays generated and assessed in Section 9.4. A counter intuitive increase in accuracy observed in coarser datasets created with larger tolerance values indicated undesirable behaviour of the variable resolution flow apportionment algorithms employed.
5. Section 9.6 investigated compression of flow accumulation arrays on real world urban surfaces. Significantly less compression was observed on complex urban faces compared to structured cone surfaces. The spatial distribution of compression was affected by differences between the scale of surface textures in modified and natural terrains.

Some of these issues (3. and 4.) relate to new algorithms presented here and have potential for rapid improvement given further development of the scaling functions and decision rules employed. Others (1. and 2.) have implications to the commonly used flat resolution geomorphometric algorithms and should inform development and application of these algorithms. Interaction between LoDs (4) did not match the analytical results well, which suggests that the conversion of the flow routing algorithms to operate in variable resolution should be reassessed at a conceptual level and learnings from these experiments incorporated.

The results of the experiments described in this chapter have suggested a road map for further research. These future research opportunities will be elaborated in the next chapter.

10. Discussion

This chapter discuses the RHSM in the context of the literature review in Part I. Section 10.1 discusses aspects of the surface model itself, and Section 10.2 discusses the evaluation of the RHSM for hydrological geomorphometry (Chapters 8 and 9). Proposed experiments and future research directions are also described in Section 10.3 and Section 10.4 respectively.

10.1. RHSM discussion

10.1.1. Cohesive framework for surface models

Geomorphometry literature tends to stress the algorithm not the surface. When surfaces are considered they are typically defined solely by reference to the classical implementations: square tessellated DEMs, node based TINs, Voronoi tessellation, or more rarely, hexagonal or unstructured tessellations (Zhang et al., 2014). Goodchild et al. (2007) decomposed spatial discretisations into geo-atoms, which are the basis of both geo-fields and geo-objects (Section 3.3). Gold (2009) explicitly recognised the common features of fields and objects in the form and nature of Voronoi diagrams (Subsection 3.4.2).

The cohesive framework detailed in Chapter 3 catalogued geo-fields by the types of structure, geometry, interpolation, and data model employed. The cohesive framework facilitates the deconstruction of algorithms. The proposed benefit of such deconstruction is to better understand the behaviour of algorithms by generalising them for different surface models and comparing their operation and outputs. For instance, see the discussion below about the affect of the geometry of tessellation on flow direction and apportionment algorithms.

10.1.2. Appropriate scale for surface models

The challenge of considering scale in distributed environmental models raised three questions in Section 5.1: Do variables change with scale? If so, how do you integrate variables

(or processes) to form a generalised value (or process)? and, if model outputs are affected by scale of input measurements and calculation scale, what scale should we use? Choosing the scale to use in a distributed model can be divided into two tasks: identifying process scale (Subsection 2.3.3) and implementing the appropriate scale in the model. The concept of process scale is well defined in literature (Blöschl and Sivapalan, 1995).

The RHSM is a structured approach to scaling and defining scale that is intrinsically process scale seeking. However, it does not rely on direct selection of model scale, instead scale selection is driven by the definition of a decision rule; the scale itself is not defined until model runtime. Therefore, scale is not an assigned parameter but an emergent characteristic of the model and the data underlying it within the constraints of the applied decision rule. Using the decision rule to define a threshold accuracy is closer to the intention of the modeller than stating a representative scale or model resolution.

The decision rule that was defined in Subsection 8.4.3 and applied in Chapter 9 finds the lower limit of variability. This is comparable to the Geographic Variance Method (Moellering and Tobler, 1972). However, only the lower limit of the scale of maximum variance is considered, the scale where variation diminishes due to uniformity of the underlying surface. The upper limit, where averaging induces uniformity, could also be considered in the decision rule, thus, identifying the peak of variability. Flow direction was chosen in Section 8.1 as the parameter to control resolution, which differs from the elevation based control employed by Vivoni et al. (2004)'s hydrological similarity TIN accumulation model.

Döll and Lehner (2002) describe up-scaling methods for grid based outputs of hydrological geomorphometry, including drainage vectors. The RHSM offers an alternative LOD approach, which involves up-scaling the inputs using a scaling rule and determining the outputs by recalculation on a variable resolution grid. This is the first time that hydrological geomorphometry has been described in a multi-resolution framework. Previous examples have been limited to up-scaling of a single parameter such as the work of Döll and Lehner (2002) and de Sousa et al. (2006).

The LOD approach employed by the RHSM to find process scale is able to support variation of process scale across the domain. This can be contrasted with Geographic Variance Methods (Moellering and Tobler, 1972), which implicitly assume constant process scale across the domain. If the process scale is very fine in some parts of the model domain, a uniform density model requires a fine scale surface over the entire domain, which may be computationally infeasible or unwarranted by the available data (Kampf and Burges, 2007).

10.1.3. Data model

The RHSM data model applies LOD modelling techniques from computer graphics to GIS operations and models. The general framework of Multi-triangulation (Puppo, 1998; Cignoni et al., 1997) is adapted in the RHSM to generate aperture 4, 7, and 9 divisions of space that are based on self-similar regular neighbourhoods. Therefore, the RHSM is effectively a splat based model that forms regular tessellations (Subsection 3.8.1).

The loss of flexibility associated with regular geometry allows simpler supporting structures and corresponding algorithms (Gerstner, 2003). de Floriani et al. (2005) identified three considerations for multi-scale models. (1) generality and flexibility, (2) efficiency (query performance), and (3) compactness. The RHSM is highly compact due to the regular implicit structure, and has excellent query performance, but is less flexible than irregular geometries.

The RHSM extends the functional surface model to define scale via the scaling function and decision rule. The error value - decision rule system provides ample flexibility to create domain specific methodologies within a common computational framework. RHSM realisations are effectively variant forms of a Quadtree (Samet, 1984). Particularly in the utilisation of hierarchical division until a level of homogeneity is reached. The key difference is that the addition of a decision rule from LOD modelling (de Floriani et al., 2005) provides flexibility in the degree of homogeneity required.

Selective refinement applies the refinements required to achieve a user defined LoD in a user defined region of interest (Danovaro et al., 2006). However, unlike a Quadtree, the RHSM is not a top down division but, instead, an analysis of scale variation. Building pyramid layers with a scaling function gives the modeller the ability to consider scale effects across several resolutions. The tiling system employed by the RHSM is similar to schemes used by Platings and Day (2004) and Bing Tile scheme (Microsoft Corporation, 2013), however, it utilises consistent indexing of values within the array and within the tile index (Subsection 6.2.2).

The RHSM is a data structure ideally suited to generating self-similar Quadtree like realisations. The flexibility provided by the scaling function and decision rule system distinguishes the RHSM from a Quadtree variant to being a scale aware spatial data structure.

10.2. Assessment discussion

10.2.1. Tessellation geometry

Consistent neighbourhoods simplify neighbourhood operations. However, surface models with regular geometries introduce geometric bias to some spatial analysis including population dispersion models (Holland et al., 2007) and hydrological flow direction algorithms (Tarboton, 1997). Geometric bias can be examined by comparing results obtained using RHSM-hex, RHSM-rec, and RHSM-tri and then adapting the algorithm informed by the observations on the different geometries.

RHSM-hex achieved greater compression than RHSM-rec (Figure 9.26), however, this was associated with increased error (Figure 9.27). This result does not support earlier work from de Sousa et al. (2006) on up-scaling drainage vectors on hexagonal grids, which saw better performance for hexagonal sampling. However, further investigation is required to determine whether the increased compression is sufficient to outweigh the increased error.

There are a number of examples in the literature of comparative studies showing that algorithms that consider more neighbours are more accurate or less sensitive to errors than those that consider fewer; For instance, Jones (1998) and Raaflaub and Collins (2006) (see Subsection 4.1.2). This effect was also seen in Section 9.2 where rectangular $D\infty$ direction was more accurate (Figure 9.6). However, hexagonal tessellation has more neighbours that are at the same distance. Six equidistant neighbours are present in hexagonal sampling, which collectively encompass the entire neighbourhood. In a rectangular tessellation, only four equidistant neighbours are available. The consistent neighbourhoods of hexagonal sampling suggests that greater use of hexagonal tessellation in science and industry would be appropriate.

10.2.2. What flow accumulation measures

Gallant and Hutchinson (2011) noted that some authors consider the direction variant cross-section of a cell when converting flow accumulation to Specific Catchment Area (a) for validation. Section 9.3 showed that correcting for cross-section does not consistently improve the performance. However, RHSM-rec appears to match Option 4 (Subsection 8.5.1), whereas RHSM-hex does not. Performing the analysis only in rectangular sampled space would have been misleading: multiple geometries must be assessed to identify geometric effects.

Another example of unnoticed geometrical biases undermining results can be seen in the analysis of the accumulation algorithm of the Mass-Flux Method (MFM) presented by

Gruber and Peckham (2009) (Subsection 4.3.2). The results of Gruber and Peckham (2009) on a cone surface appear to conform to the expected concentric circles. However, because the study area is square, greater accumulation should be seen flowing from the corners[1]. The error generated by the geometric bias of the corner neighbours would become apparent if the domain was restricted to a circle as was done in Subsection 9.1.2.

10.2.3. Accumulation algorithms

Single and multiple direction algorithms

$D6$ flow direction and accumulation had been described previously in de Sousa et al. (2006) and Wright et al. (2014). Subsection 8.3.1 defined a generalised single flow direction as Dx. In addition, Subsection 8.3.2 generalised the $D\infty$ algorithm of Tarboton (1997) to any geometry of tessellation by utilising the cohesive framework for surface models described in Chapter 3. The generalised algorithm gave insight into the behaviour of the apportionment algorithms not apparent in the original geometries. For instance, $D\infty$ performs badly when flow direction is evenly distributed between neighbouring facets as indicated by the anisotropy in Figure 9.14.

Pan et al. (2004) and Erskine et al. (2006) distinguish single and multiple direction algorithms and find that multiple flow direction algorithms are perferred on some surface types. Relative differences were greatest along ridges and side slopes, and differences decreased where the terrain became more convergent. Erskine et al. (2006) recommend Multiple-direction algorithms on undulating terrains because they allow for flow divergence. However, these studies do not separate the direction and accumulation components. For instance, Pan et al. (2004) identifies $D\infty$ as a bi-flow algorithm, however, this is only true of the accumulation; the direction component of $D\infty$ is singular. Single direction algorithms do not readily model dispersion contributing to the observed anisotropy seen in Figure 9.14.

Apportionment forms

Orlandini et al. (2003) described two methods that carry discrepancies of flow direction from the $D8$ method along the flow path known as $D8$-LAD (eight drainage directions, least angular deviation) and $D8$-LTD (eight drainage directions, least transverse deviation). Tarboton (1997)'s $D\infty$ algorithm used angular displacement. Orlandini and Moretti (2009) identify the difference between the two regarding relative apportionment

[1]It should be noted that Gruber and Peckham (2009) describe this as a preliminary analysis.

near the facet edges but indicates that it is not clear which is more effective. Section 9.3 introduces a third alternative, areal displacement, and compares the three for the dispersive and concentrative cone surfaces. Figure 9.19 showed an overall better performance of linear. This is a useful conclusion, although, it is necessary to confirm this better performance on other surfaces.

There are examples in literature of apportionment algorithms that are adjustable. For instance, MFA can be distributed proportional to slope (Quinn et al., 1991) or proportional to slope to an exponent (Freeman, 1991; Quinn et al., 1995; Seibert and McGlynn, 2007). Adjusting the exponent fine tunes the behaviour of multiple MFA algorithms for specific surfaces. It may also be beneficial to adjust apportionment within the RHSM for different surfaces.

10.3. Proposed experiments

10.3.1. Scale sensitivity

In a model that varies with scale, scale is effectively a parameter and a candidate for sensitivity analysis. Given the treatment of scale as a variable, experiments are required to investigate how modellers can achieve the sensitivity analysis, optimisation, and validation for scale. In the RHSM, scale is expressed in the decision rule, therefore, the experiments described in Section 9.5 are effectively a form of sensitivity analysis of scale in hydrological geomorphometry.

The scale sensitivity analysis of hydrological geomorphometry should be extended by analysing how the model output is affected by changing the tolerance and, therefore, resolution of other parameters in the model; For example, with elevation or accumulation as the resolution driving parameter. This experiment would determine whether variable realisations based on direction preserve hydrological accuracy better than the equivalent elevation compression.

It would also be interesting to repeat the experiment with direction compressed but calculating accumulation at the reference scale. This would provide insight into how compressing direction affects accumulation without the additional and significant effects from variable resolution flow accumulation being present. The results would also be useful as a benchmark for assessing alternative variable density accumulation algorithms. Another experiment was described in Subsection 9.4.4, which suggested calculating flow direction independently on each level of the RHSM pyramid and then devising a decision rule that identifies appropriate scale by comparing between levels.

10.3.2. Error noise complexity

The RHSM should be used to investigate the sensitivity of hydrological geomorphometry to noise. The proposed experiment would be similar to Monte-Carlo error analysis but the goal is not to quantify error but, instead, to observe scale effects. The experiment would generate an artificial surface either a linear slope or simple polynomial and then add controlled noise to the data and examine how the models change as variability changes. The experimental results would allow production of statistical models for a range of surface complexities that could be related to surfaces found in various environments including urban ones.

The above analysis would extend the work of Endreny and Wood (2001) by introducing multi-scale surface models into the analysis of the sensitivity of flow routing methods to elevation errors. Another effect that may be relevant to scale is the observation of Oksanen and Sarjakoski (2005) that errors increase as noise correlation distance increases. An understanding of the effect of scale on errors may explain this effect.

10.3.3. Hydrological geomorphometry

Some extensions to the hydrological geomorphometry experiments are described here.

Catchment boundaries

The RHSM Hydrological geomorphometry described in previous chapters can be extended to align with existing single scale modelling practises by, for instance, introducing time, subsurface hydrology, slope, topographic wetness index, and catchment delineation. Storing a value for threshold draining that can be adjusted up or down to merge catchments without rerunning the analysis is an interesting idea (Liu and Snoeyink, 2005). Catchment boundaries require non-dispersive accumulations for instance Orlandini and Moretti (2009)'s $D8$-LTD method.

de St. Venant

There are two ways that the RHSM could be used for de St. Venant flow modelling.

1. The RHSM could be used to define variable density meshes for existing numerical solution models to the de St. Venant equations, such as HEC-RAS (Tate et al., 2002).

2. The de St. Venant equations could, potentially, be solved within the RHSM structure itself. It also may be beneficial to use a variation of Morris and Woolhiser (1980)'s

guidelines for when the kinematic equation is valid to control the type of solution utilised by the model as an extension of the function of the decision rule.

Flow direction methods

Flow direction algorithms can be investigated in a number of ways in light of the analysis in this thesis. For instance, the modifications to the Rho8 flow direction algorithm suggested in Subsection 8.3.3 are an opportunity to advance the work of Fairfield and Leymarie (1991) and Wright and Leonard (2012). The analysis of Dx and $D\infty$ flow direction and accumulation algorithms could be extended by adapting more algorithms from the literature (see Section 4.3) to the RHSM and including them in the assessment. The results from the direction analysis in Section 9.4 would be clearer if plotted against compression achieved instead of level. A more experimental possibility is to apply the RHSM to the Modifiable Area Unit problem mentioned in Subsection 9.4.4.

The problems Section 9.3 identified in the $D\infty$ algorithm may be addressed by creating a two facet (i.e. three neighbour) form of the $D\infty$ direction algorithm. Such an algorithm could consider whether the three cells were concentrative or dispersive and adjust the apportionment accordingly. Alternately, the RHSM could utilise the hierarchical structure to determine direction using finer or coarser relatives.

Apportionment algorithms

It may be possible to fine tune $D\infty$ further by adapting the apportionment algorithm. This could be accomplished by further analysis of the differences identified between geometries and scales, and/or by analysis of the conceptual and mathematical models. The demonstrated shortcomings of the flow accumulation methods investigated provide pointers to possible improvements. For instance, an apportionment algorithm that allows flow to more than two neighbours may reduce anisotropy.

An alternative approach would be to optimise the apportionment algorithms using a stochastic method to try numerous possible apportionment algorithms to identify, empirically, generally applicable solutions. The most successful formulations could then be analysed to determine the physical basis of their suitability.

The variable resolution flow accumulation algorithm needs to be redesigned to address the weaknesses identified in Section 9.5. An initial exercise is to implement and assess the other models for variable density accumulation described in Subsection 8.5.3.

10.3.4. Urban surfaces

Not withstanding the necessary improvements to the variable resolution flow accumulation model, progress toward an urban RHSM urban run-off model could be made by undertaking the following experiments.

Flow routing in urban environments

The Dx and $D\infty$ experiments described in Chapter 9 for mathematical surfaces can be conducted using an urban surface model. However, this would require a definition of true direction and accumulation. A preliminary approach would be to construct a surface that has urban characteristics out of differentiable mathematical shapes. A surface that includes buildings, roads, and channels could be constructed from planes and simple curves.

A surface as simple as a plane with a channel would shed light on the affect of geometry and direction anisotropy in the generation of sinks. It is not currently clear whether the isotropic neighbourhood of RHSM-hex would reduce formation of sinks compared to RHSM-rec or whether the fewer neighbours of RHSM-hex would increase false sink formation.

An urban surface developed from mathematical shapes would be highly idealised. However, to perform the experiments on an urban surface model constructed from LiDAR or photogrammetry would require a definition of truth for the study domain. Comparison data to validate urban hydrological models are rare and at least expensive if not infeasible to collect. However, some data do exist, for example regional councils in New Zealand develop 1, 2, and 3-D models of urban areas for flood risk assessment, which can be used to compare flow predictions.

Rather than comparing between different models, it would be better to compare models to observations. Reported observed urban flood pathways and flood extent are present in academic and commercial literature. Other potential sources of validation observations are citizen reports (Gaitan et al., 2015), and remote sensing datasets that distinguish wet and dry areas during flood events and may be observed by drone surveys (Schumann et al., 2009). A final option is urban flooding created in controlled lab conditions.

Hydrological conditioning and LOD modelling

There is an important interplay between hydrological conditioning and LOD modelling. Hydrological conditioning will affect the results of LOD modelling and the decision rule

can be used as a form of hydrological conditioning. Just as interpolations for hydrology should consider drainage structure, the decision rule and error value in LOD modelling should consider drainage structure. Further experiments could be performed to investigate this relationship.

The intention of the model developer is crucial in the selection of hydrological conditioning approaches and the LOD decision rule so that appropriate scale can be determined without manual assessment of individual areas. If, for instance, the intention of the modeller is the construction of a flow direction surface for flood modelling of urban areas, the preservation of high resolution around buildings may be justified by the high value that residents place on whether those areas are likely to be affected by flooding.

Compression of flow direction arrays would benefit from conventional hydrological conditioning techniques such as smoothing, pit filling, and introduction of ancillary data (Section 4.4). Smoothing base data before LOD modelling would increase compression at the cost of potentially obscuring important variation. Filling sinks would also create flat areas, which would be suitable for compression.

LOD modelling introduces some alternative approaches to hydrological conditioning. For instance, smoothing in an LOD model need not be performed on the surface; instead, the error value or decision rule could be altered by ignoring outliers when deriving the error value. For flow direction modelling, slope could be introduced to the decision rule so that differences between child directions on relatively flat surfaces are given less significance, effectively increasing the tolerance in flat areas. Decreasing the threshold value of the decision rule with increased accumulation my help resolve the critical part of the flow network

In an LOD model, known surface features could be introduced as ancillary data in vector form before compression to ensure that these features are preserved. Indeed, densification of the sparse model to greater than base level resolution in areas of known hydrological features could be desirable: effectively allowing ancillary data to affect resolution. Using fractional HIP addresses in areas that contain important features is one way to achieve local finer resolution, whether for sink resolution or ancillary data. In a sparse LOD model, this could be implemented by the insertion of channels that flow through cells that are smaller than the base resolution, such a technique would be in accordance with Lindsay and Creed (2005a)'s objective of least impact sink resolution.

10.4. Proposed RHSM improvements

Recommendations for further work to improve the RHSM are given below.

10.4.1. Efficiency and development of RHSM computational model

The computational model was developed as a proof of principle implementation. Before further development is applied to the computational model, it would be advisable to rebuild the code in a more readable and modular form that is simplified to the essential elements. Assuming that development continues in a non-commercial open-source development model, such work should be hosted on an open source platform.

In order to become a usable platform, the performance speed of the RHSM computational model must be improved. Reassuringly, many improvements of the RHSM computational model are possible that have not yet been implemented.

1. Implementation in a more efficient language i.e. a C variant.
2. Implementation of multi-core processing.
3. Utilisation of graphics card processing.
4. Improvement of visualisation speed by implementing a vector tile format (supported by ESRI and Mapbox), which could be viewed by a browser.
5. Improvement of efficiency of algorithms.

One high value target for more efficient implementation is the addition function. More generally, if many local neighbourhood operations will be performed on a dataset, it would be very efficient to store the neighbours in the form of neighbour arrays rather than calculating neighbours on the fly using addition. Finding neighbours is very easy for cells with a HIP ordinate ending in 0 (centre cells of agglomerations). Algorithmically shifting the centre to create views of the array from which the neighbours of other cells can be extracted as easily as the centre cell would markedly improve neighbourhood operation speed.

A very simple improvement, that was not implemented, is the application of visible scale ranges so that fine resolution data are not displayed when zoomed out. A more radical efficiency measure would be to adopt the model of raster functions implemented in ArcGIS Pro to develop real time processing of subsets of an RHSM.

10.4.2. Improvement of mathematical model

Some necessary additions to the mathematical model are given below.

1. An efficient method to calculate HIP ordinates given (x, y) coordinates is missing. One potential solution is to adapt the Morton address to hexagonal axis.

2. Error value calculation could represent the variation using a fractal variable or a measure from the variogram.
3. Add support for No Data cells to the Condat interpolation.
4. More interpolation algorithms

The interpolation methods implemented in the RHSM were listed in Subsection 7.3.6. These interpolation methods create artefacts around sudden changes of elevation and may alter drainage patterns. Hydrological applications rely on surface shape and drainage networks rather than elevation directly. Therefore, interpolation models that consider shape are required. However, elevation based methods such as kriging and second order least squares may be important for other applications.

Some literature examples of interpolations that consider drainage structure are Mass Balance (Grimaldi et al., 2005), ANUDEM (Hutchinson, 1989), Regression kriging (Hengl et al., 2008), and network snakes (Goepfert and Rottensteiner, 2009). These methods also permit the use of auxiliary predictors, such as river lines, in the interpolation.

10.4.3. Further development of the RHSM conceptual model

Some refinements to the RHSM conceptual model that are worth considering are,

1. Application to a global referencing frame.
2. Introducing fractional HIP addresses to provide finer than base scale resolution in restricted areas, if such data are available.
3. Relaxing the hierarchy geometry to allow a greater range of resolutions.
4. Introducing more flexible hierarchy branching sites. One significant conceptual change in the model structure would be to allow branching at fractional scale ratios (in addition to the fractional spatial addresses proposed in refinement 2.). How this would be accomplished computationally is not clear but it would permit analogue scale ranges.

10.4.4. Other applications of the RHSM

Significant unrealised potential exists for using the RHSM to improve geomorphometric algorithms by comparing the results of operations conducted with the different geometries. It is currently unknown how some aspects of geomorphometric analysis, whether hydrological or otherwise, are affected by the geometry of tessellation. RHSMs could be utilised in experiments to determine whether the geometry of tessellation affects pit

formation, extent of hydrological conditioning required, and sensitivity to errors in data; to mention just a few examples from hydrology (see Section 10.3). Such studies would benefit from inclusion of irregular Voronoi tessellations.

There is also unexplored potential to improve understanding of scaling by using the RHSM as a scaffold to describe and apply scaling functions and investigating how the results compare to measured observations, if such variant density observations exist. Obtaining such test data would be difficult given the rigidity of the multi-scale structure. Different scale observations could be sourced from ground, aerial and space platforms but these platforms would not necessarily be aligned with RHSM levels, which is further impetus to develop more a flexible multi-scale geometry.

This chapter discussed the RHSM and its assessment in the context of relevant literature. What follows will conclude the thesis in the context of the objectives of the research.

11. Conclusion

11.1. Summary

This section summarises the thesis in relation to the three core motivations and five research questions that were stated in Section 1.3.

11.1.1. Surface models for hydrological geomorphometry

The first objective was to identify the characteristics of a surface model that considers scale and geometry in geomorphometric catchment analysis. This objective was captured by two research research questions: 1. What are the characteristics of GIS surface models and how can they be catalogued and understood? And 2. What kind of surface model provides the ability to model hydrological processes that occur at different scales in different areas? This objective was primarily addressed in Part I.

Part I explored a diverse selection of academic literature with the goal of finding ideas that could be developed into a new surface model that is optimised for geomorphometric hydrological modelling. In Chapter 2, after a look at the nature, benefits, and limitations of models in general and hydrological models in specific, the complexity of modelling distributed variables that vary with scale emerged as a core motivation in the development of a surface model for hydrological modelling.

Chapter 3 developed a systematic framework to describe and catalogue the various surface models in use. Two themes that emerged from this process were the potential for hexagonal tessellations to be better utilised for spatial problem solving and the growing importance of hierarchical data structures for describing and computing 3-D space. A conceptual intersection of ideas between scale and hierarchy was discovered that became the inspiration for the RHSM developed in Part II.

Chapter 4 turned to the various techniques of parametising surfaces numerically, collectively known as geomorphometry. The main focus was a multi-step process of static flow routing, which was adapted for hierarchical surface models and assessed in Part III. The

systematic description of surface models in Chapter 3 proved to be a useful tool to deconstruct geomorphometric algorithms and reconstruct them in generalised form that can operate on different surface models.

11.1.2. The proposed surface model

The second objective was to identify a surface model that meets these criteria and develop a mathematical and computational framework to integrate the proposed surface into a GIS. This objective was encapsulated by research question 3: How can a surface model that models processes at different scales in different areas be implemented?

Part II set out the RHSM, which is the proposed surface model. The conceptual motivation was to model parameter variability with scale explicitly using a hierarchical data structure. This was achieved by implementing a regular discrete structure for both the spatial dimension and the scale dimension in the form of a regular pyramid. The regular structure was compatible with implicit data structures. The surface model utilised an adapted version of the HIP addressing system of Middleton and Sivaswamy (2001) that was generalised for the regular tessellations: triangular, hexagonal, and rectangular.

Borrowing ideas from variable resolution data structures for computer graphics, a framework was developed to identify appropriate resolution for modelling based on the concept of Level of Detail (LOD) modelling. The RHSM combined pyramid layers generated by a scaling rule with error values and a decision rule to generate variable density realisations.

11.1.3. Hydrological Applications

The third objective was to evaluate the performance of the new surface for geomorphometric catchment analysis with respect to scale and geometry.

Part III built on the RHSM by developing and evaluating a set of hydrological applications with the intention of performing geomorphometric catchment analysis on multiple scales and capturing complex surfaces without resorting to excessive resolution or manual determination of appropriate scale. Chapter 8 proposed variants of $D\infty$ and Dx flow routing algorithms that were generalised for various geometries, and methods to perform accumulation on variable resolution realisations generated from flow direction arrays using simple scaling functions, error values, and decision rules. These developments affirm research question 4: Can the algorithms of flow accumulation and direction be generalised to hexagonal and variable resolution surfaces? Chapter 8 also introduced a methodology to adjust the apportionment of flow accumulation algorithms; Linear, Angular, and Areal forms of $D\infty$ apportionment were described and implemented.

Chapter 9 described an assessment of the RHSM flow routing procedure that solved the algorithms on mathematically defined cone surfaces and compared the results to analytically defined solutions. The RHSM highlighted the interaction of geometry and scale with the algorithms. These findings were summarised and discussed in Section 9.7 and addressed research question 5: How does the surface approximation in geomorphometric catchment analysis affect model outputs?

The long term goal of this research is stated in research question 6: Can we resolve complex overland drainage networks using multi-scale surface modelling? Although promising indications were seen in this research, this question cannot yet be answered in the affirmative.

11.2. Conclusions

One of the most beneficial and interesting aspects of this thesis was the deconstruction of flow algorithms into components that can be adapted and reassembled in hexagonal and rectangular space. This process of deconstruction and generalisation to different geometries provided a deep understanding of the operation of the algorithms. Three specific notable findings were:

1. That flow accumulation arrays should be compared to specific catchment area using the cell size (one dimensional density measure) as a substitute for contour length rather than considering the cross-sectional length of the cell as it varies with direction.
2. That linear was the overall best performing apportionment algorithm for $D\infty$.
3. RHSM-hex achieved better performance for compressing flow direction arrays.

Application of the methodology to urban environments was restricted by limitations identified in both the underlying flow accumulation algorithms and the LOD method trialled. The implementation was very efficient for generating pyramids. However, for some other tasks it was less effective. For instance, converting the outputs into a form that can be viewed in the ArcGIS platform was slow. The issues identified can be addressed using the recommendations in Subsection 10.4.1 to begin developing real world models.

Notwithstanding the challenges that remain to be overcome, RHSM sparse realisations offer a formal data structure to capture, within distributed environmental models, fine details with large effects. The approach described in this thesis departs from methods to achieve similar results via altering the distribution density of numerical solution schemes. The LOD process used to generate data dependent sparse realisations encourages the user

to focus on selecting appropriate decision rules and scaling functions rather than manually defining the resolution for specific areas.

Scale and geometry are primary considerations for modellers that are translating the analogue complexity of the real world into digital domains for geographical modelling. This thesis, motivated by pragmatic simplicity and implicit efficiency, has proposed a regular structure for both scale and geometry. The usability and applicability of the model must follow. The strengths and weaknesses of implicit regularity of scale and geometry in surface models complement more flexible irregular explicit schemes.

The processes outlined in this thesis represent a new and promising method for analysing complex surfaces. The vision that motivates the model presented here is a user friendly GIS tool that modellers can use to implement scaling rules and decision rules to perform a wide range of multi-scale analyses, where the scaling and decision rules represent best practise understanding of the processes being modelled. This goal is achievable now that the fundamental concepts have been demonstrated.

It is imperative for communities in our growing towns and cities to balance our desire to shape the environment to our needs with the necessity to find an enduring way to live within the constraints of the environment. Effective management of stormwater requires the full and appropriate use of the data streams that our spatial technologies and information processing abilities allow; variable scale modelling in GIS is a critical component of this endeavour. Further research into the RHSMs introduced in this thesis will improve spatial models by explicitly recognising and modelling the effects of scale and geometry.

Appendices

A. Model quality and assessment

In this thesis the term "model" was used to describe a digital representation of a surface and also to describe a digital representation of a environmental process. This appendix discusses assessment of environmental models, see Subsection 3.2.3 for assessment of Digital Elevation Models. For further reading, see a collection of papers edited by Anderson and Bates (2001).

Hydrological models are approximations of reality that contain assumptions and simplifications. However, models are often expected to be useful in making predictions in situations that have not yet occurred or where measurements have not yet been made (Beven, 2001). It is reasonable to want to know how accurate such predictions would be and how effective a model is at representing the behaviour of the system it describes. However, to prove conclusively that a model is completely accurate, one would need to completely understand the real world phenomena that is being modelled, for natural and engineered systems this is difficult task.

Due to the inherent limitations of modelling and the difficulty of accurately assessing the validity of hydrological models, catchment hydrology has been referred to by some as a trans-scientific discipline one with "questions that can be asked of science and yet cannot be answered by science" (Beven, 2001). The challenge of accurately assessing hydrological models makes rigorous testing and assessment of models even more important. Model assessment involves several interrelated components, including documentation, verification, error budget, error propagation, sensitivity analysis, optimisation, and validation. The components of model assessment are discussed below. Following these procedures will not prove that the model is accurate but will provide a measure of quality assurance.

A.1. Documentation

Perhaps the easiest but also among the most essential components of model assessment is comprehensive documentation of the conceptual, mathematical, and computational stages of model development. Model documentation and testing are crucial to ensure that results

are reproducible and scientifically sound (Sasowsky, 2006). It is very important that the model developer documents the limitations of the model and clearly articulates them so users apply the models correctly.

Oreskes and Belitz (2001) provide an insightful discussion of the philosophical issues of model assessment including a warning that "the modeller may be blamed for faulty predictions caused by limitations of which he or she was well aware, and perhaps even warned people". Ironically Oreskes and Belitz (2001) inadvertently provide an example. Oreskes and Belitz (2001) claim that the controversial Limits to Growth (LtG) study (Meadows et al., 1972) inaccurately predicted "wide-spread natural resource shortages, exponential price increases for raw materials, and possibly global economic collapse before the end of the century." However, Meadows et al. (1972) made no such claim.[1]

That both the authors of Oreskes and Belitz (2001) and the paper's reviewers overlooked or ignored misstatements of fact regarding the findings of Meadows et al. (1972) indicates cognitive bias; in this case a negative one toward the LtG model. However, in other situations involving models that have been developed by recognised experts and used frequently in the past, there is a very real danger that the models will be applied erroneously with overconfidence that the conceptual model is sound as it has been developed by clever people who are the best in their fields. Clear documentation of the limitations, assumptions, and methods employed in the model is the best defence against both unfair criticism and unsuitable application due to overconfidence.

[1] Oreskes and Belitz (2001) note that the end of the century has come and gone without these events occurring and suggest that one of the reasons for the failed prediction of the LtG study was that the authors of the LtG study made a conceptual error regarding the definition of resources that caused them to underestimate available non-renewable resources. This critique is flawed in three ways. 1) The LtG did not include any predictions and was not intended as a predictive model. Rather, it presents an ordered set of assumptions that indicate modes of behaviour of the world system and allows the response of the system to different policies to be tested. 2) The "standard run" model, the most pessimistic of the scenarios investigated by the LtG study, indicates peak resource use in the middle of the 21st century not the end of the 20th. 3) Meadows et al. (1972) were aware of their ignorance of the true quantity of resources. They used an optimistic estimate for resources and included a parameter that allowed them to vary total resources to investigate how the model behaved under different assumptions. Both doubling resource and infinite resource assumptions were tested. In both cases the model led to a collapse in industrial output caused by pollution. In 2017, the LtG model as a predictive model is neither validated nor invalidated. Recent reports suggest the advice to drastically reduce consumption of non-renewable resources is still sound (IPCC, 2014). Subsequent editions of the limits to growth book have sought to make these points more clear (Meadows et al., 2004).

A.2. Verification

Accurate documentation should be combined with thorough testing of the model to verify that it is performing in accordance to the specifications. Ensuring that a model adheres to specification is sometimes called verification and should not be confused with validation, which is assessing if a model adheres to real world behaviour i.e. the system it is representing (Barnsley, 2007). Code frequently contains unintentional errors, which affect model outcomes. Thorough testing is essential to locate and remove such bugs and ensure that a model performs calculations as intended. Roerdink and Meijster (2000) warn to be wary of modifications and optimizations of the computational model, which can lead to departures from the algorithm specification.

A.3. Error budget

It is necessary to identify and account for the sources of errors in a model and where possible to quantify the effects of these errors on model outcomes. Errors can be assessed a priori by analysing the inputs and structure of the model, and a posteriori by comparing the model outputs with observations or by the internal consistency of the data within the model framework.

There are three broad categories of error, data errors, model conceptual errors, and numerical errors. Data errors may include input data, uncertainty or errors in the data used to parameterise the model, and errors in the data used to validate the model. Conceptual errors may including simplifications and assumptions made in the conceptual model, errors or inappropriate methods employed in the mathematical model, or mistakes or incorrect techniques in the implementation of the computational model. Numerical errors in the computational model may include limitations of numerical solution schemes, inexact arithmetic, or software and hardware numerical precision limits (Barnsley, 2007). Numerical errors can be significant in hydrological models, especially time stepping models. Such errors can be reduced by using error minimizing variable time steps (Clark and Kavetski, 2010; Kavetski and Clark, 2010). Given the sources of uncertainty, it is appropriate to produce an error budget of the known sources of error in the model and to assign likely values to these errors.

A.4. Error propagation

Errors should be propagated through the model to see the range of outputs possible given known uncertainties from model, data, and parameter uncertainty. Errors can be propagated analytically or stochastically. Heuvelink (1998) describes methods for propagating errors in environmental models. Analytic techniques are more suited to local parameters such as slope and aspect rather than regional parameters such as flow accumulation (Oksanen and Sarjakoski, 2005). Stochastic error propagation can be performed using Monte-Carlo analysis (Endreny and Wood, 2001; Raaflaub and Collins, 2006; Lindsay and Evans, 2008; Heuvelink, 1998). Gonçalves and Santos (2005) demonstrate interval arithmetic as an alternative to Monte-Carlo for slope classes.

Monte-Carlo involves generating realisations of your dataset by adding randomly generated errors to the data or by producing sets of parameters which vary within a realistic range. Running the model a sufficient number of times with different realisations generates a range of plausible outcomes, which can be analysed to determine the effect of data or parameter errors. Van Niel and Laffan (2003) describe techniques to determine how many realizations are required for Monte-Carlo error propagation. The outputs of Monte-Carlo analysis can be averaged to create probability weighted flow tubes (Endreny and Wood, 2001), or probability weighted catchments (Wright and Leonard, 2012).

Data errors can be auto-correlated or uncorrelated (Raaflaub and Collins, 2006). Several authors have investigated the effect of error correlation on surface hydrological models. Endreny and Wood (2001) showed correlation was not important for error propagation on run-off flow paths. However, careful study of error correlations on slope, aspect, and catchment found errors increased as spatial correlation range increased, contrary to the prevailing assumption that no auto-correlation is the worst case scenario (Oksanen and Sarjakoski, 2005). In addition to the auto-correlation model, it is important to be aware of biases in random number generators and to select the appropriate ones because they have been shown to affect the result of Monte-Carlo analysis (Van Niel and Laffan, 2003).

A.5. Sensitivity analysis

Sensitivity analysis is the study of the effect initial values of input parameters have on the outputs. Sensitivity analysis perturbs one or more of the initial parameters to investigate the effect on the result of the model. Sensitivity analysis differs from error analysis because the goal is to understand the relative importance of individual parameters on the output. See Ros and Borga (1997) and Francos et al. (2003) for sensitivity analysis of hydrological

models. Lindsay and Evans (2008) showed that various morphometric properties derived from DEMs are unstable given likely error ranges.

Sensitivity analysis can identify model inputs that cause significant uncertainty in the output and are therefore candidates for further research to improve the model. Alternatively, parameters that have no effect on the output may indicate redundant parts of the model structure, which could be removed.

The one parameter at a time method of performing sensitivity analysis does not capture interactions between parameters and, therefore, becomes increasingly inaccurate as parameter numbers increase (Saltelli et al., 2000). Distributed models support different parameters in each cell over the entire spatial discretisation, giving rise to a great many parameters and a consequential challenge to understand parameter sensitivity. As discussed in Kampf and Burges (2007), in many models, there is a trade off between comprehensive process representation and over-parametrisation. Ockham's razor applies, which states that if a simple model will suffice, none more complex is necessary. A counter argument to the principle of Ockham's razor in hydrological models is that nuance and subjectivity are required to represent our best understanding of complex systems.

A.6. Optimisation

Optimisation is the tuning of a model to produce desired outcomes in a specific domain by altering the values of various parameters within reasonable ranges. Sets of model parameters that produce the desired outcome are considered behavioural. If the conditions that prevailed during the calibration period were unusual the optimisation may be unsuitable in other domains. There are several methods to identify optimal parameter sets including the University of Arizona stochastic methodology called shuffled complex evolution, (UASCE) (Duan et al., 1992), simulated annealing, genetic algorithms, Monte Carlo Markov Chain methodologies (Kuczera, 1997), and multi-objective optimisation techniques such as the Pareto optimal set methodology (Madsen, 2003; Gupta et al., 1998).

The more parameters are present in a model the more flexible the model output becomes. In an over-parametised model there is a large range of parameter combinations that leads to results that match the desired results. If there are multiple parameters sets that produce the desired outcome this is non-uniqueness. Oreskes and Belitz (2001) identify three types of non uniqueness, numerical (more than one solution to the governing equations), parametric (multiple combinations of effective parameters), and conceptual (multiple conceptual models that can account for the empirical evidence). Beven (2001)

argues that the existence of multiple sets of parameters (and indeed multiple conceptual models) that produce the desired results is not necessarily evidence of a poorly designed model but may, in fact, be a fair representation of the level of knowledge of the system. Beven (2001) calls this non-uniqueness as a final outcome "equifinality".

The value of a parameter that causes the model behaviour to match real world behaviour will change depending on the values of other parameters in the model (Beven, 2001). It is impractical to optimise all these parameters and parameter combinations. Therefore, general case estimates are used. Complex models may contain more than one error. If errors cancel each other out, the model may fit the test data but be a poor predictor in other areas (Oreskes and Belitz, 2001). The fidelity of the conceptual model can be verified by how well parameter tuning transfers between domains.

Beven (2001) and Beven et al. (2000) developed the Generalised Likelihood Uncertainty Estimation (GLUE) methodology to accommodate the fact that it is not possible to identify an optimal set of model parameters. GLUE is an extension of Generalised Sensitivity Analysis (Spear et al., 1994) in which many different model parameter sets are chosen randomly, simulations run, and evaluation measures used to reject some model structure/parameter set combinations as non-behavioural, while all those considered as behavioural are retained in prediction. In GLUE, the predictions of the behavioural models are weighted by a likelihood measure based on past performance to form a cumulative weighted distribution of any predicted variable of interest.

A.7. Validation

Given a model that has been documented and verified to demonstrate that the simulations are producing results in accordance with the specifications; and the known sources of error have, to the extent possible, been quantified and propagated to model outputs; it is then possible and necessary to assess the model against its intended purpose. This is validation, a critical assessment of the model to verify that both the model outputs and its internal states are close to those measured in the real system.

Whether a model is valid may depend on the individual expectation of what a valid model should accomplish, which depends on subjective interpretation. Oreskes and Belitz (2001) argue that validation of hydrological models is difficult due to non-uniqueness (equifinality as discussed above), temporal and spatial divergence, and subjectivity of model assessment. The extent of the data available for validation may be very limited compared to the period or area to which the model will be applied. The scale at which

validation data was measured and recorded may not be commensurate with the scale of the model.

Validation is ideally accomplished by comparing results against independent lines of evidence. In the case of hydrology, this involves field measurements, bearing in mind parameters may have been tuned to these measurements. Unfortunately the geographical and chronological scales of hydrological processes, and the challenge of access to the subsurface make it difficult to find comprehensive independent evidence with which to test hydrological model claims (Oreskes and Belitz, 2001).

Models can be assessed in their entirety or piecewise. Assessing the entire model is important because the sum of the errors may be greater than the errors of the parts. However, errors apparent in the behaviour of the whole model may be the result of many things and the correct approach to address the errors, other than by arbitrarily tweaking parameters, may not be obvious. The whole model can be assessed by prediction of another period of data (a split-record test). A stronger test would be an independent check of the predicted internal states of the system. This is also problematic, however, since most internal variables have to be measured at scales much smaller than the grid or catchment scales of the model predictions (Beven, 1996).

Where independent lines of evidence are unavailable, alternatives need to be found. Assessment can be accomplished by comparing between models, either with a different model, or a similar model with greater resolution, or with a more computationally intensive model (Raaflaub and Collins, 2006). Comparing between models is particularly helpful for assessing the effect of assumptions made in the conceptual model. However, great care needs to be taken in assessing which model is more correct based on inter model comparisons. Another alternative is to compare the proposed model with established methods, which represent best practice.

Whether due to the finer spatial resolution or due to the more detailed calculations required, comparisons with a higher fidelity models are necessarily performed on smaller domains with the results hopefully being applicable to wider areas. If the high fidelity approach is plausible over the entire domain, it would be sensible to use the high fidelity model rather than employing it is a check.

Models can be tested piecewise in domains where components of the model can be determined analytically, For instance. surface flow algorithms can be tested on basic mathematical surfaces, such as, inclined planes, convergent and divergent cones, and more complex example such as Morrison's Surface III (Jones, 1998).

The statistical measure of accuracy deployed for model assessment depends on the type of model output being assessed. For an introduction to spatial statistics see Cressie (1991).

Some statistical measures of the "goodness-of-fit" between the model and observed values are listed here: modelling efficiency statistic (Döll and Lehner, 2002), coefficient of determination (r^2), or chi-squared (χ^2) (Barnsley, 2007), paired student t tests, Kolmogorov-Smirnov (k-s), standard deviation, Pearson product moment correlation coefficients, Wilcoxon sum-ranks test, (Wang and Yin, 1998), variograms, histogram, and semi-variogram model (Holmes et al., 2000). Statistical techniques for validating surfaces are discussed in Subsection 3.2.3.

B. Urban Hydrology

This appendix describes the hydrological cycle and summarises urban hydrology. General overviews of mathematical hydrological modelling can be found in Singh and Woolhiser (2002) and Kampf and Burges (2007), or Fletcher et al. (2013) for urban hydrological modelling.

B.1. The hydrological cycle

Powered by energy from the sun and the gravitational pull of the Earth, water is driven through a large, complex, and never ending cycle, which encompasses the oceans and atmosphere, as well as the surface and subsurface of the Earth. This cycle is known as the hydrological cycle and is the focus of the science of hydrology (Chow et al., 1988). The hydrological cycle can be subdivided into three domains: atmospheric, subsurface, and surface. A simplified schematic of the hydrological cycle is depicted occurring over land in Figure B.1.

The dominant hydrological processes in the atmosphere are evaporation, atmospheric circulation, and precipitation, which all occur in the troposphere. Atmospheric circulation is driven by the heat difference between the poles and the equator, and by the rotation of the Earth (Chow et al., 1988). Atmospheric circulation can drive water from the oceans far inland. Within the troposphere, temperature decreases with altitude. Therefore, when atmospheric circulation causes air to rise, water vapour may condense around condensation nuclei to from droplets, or if the temperature is below freezing, ice crystals, which may coalesce to form snow. Droplets grow through collisions and condensation, and shrink due to evaporation, causing them to rise and fall within clouds until the weight of the droplet causes gravity to overcome friction and the droplet breaks free, to fall as precipitation in the form of rain, snow, hail, or sleet. The quantity of water reaching the land surface via precipitation varies significantly from place to place and over time (Chow et al., 1988).

Upon reaching the ground, water is stored or driven by gravity on a path back to the oceans. Water that has reached the ground in solid form may melt shortly after or spend

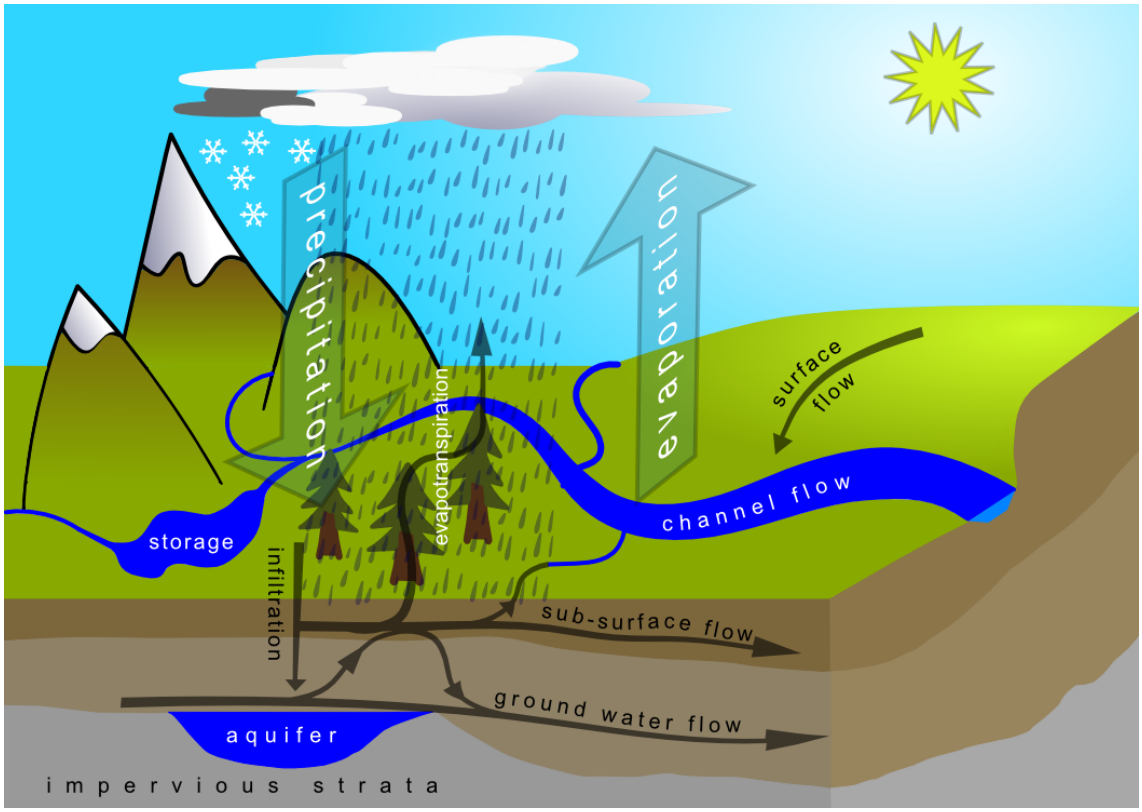


Figure B.1.: The hydrological cycle.

a considerable time as ice in glaciers or ice caps. Porous soil and rock strata contain voids that permit water flow; other impermeable rock and soil strata do not. Water falling over land as rain, or water from melting snow and ice may infiltrate the upper levels of soil and become subsurface flow or may infiltrate more deeply into the ground and become ground water flow. Subsurface flow occurs in the unsaturated or vadose zone where the voids contain some air. Ground water flow occurs in the saturated zone where the voids are entirely full of water. The water table is located where the saturated zone is at atmospheric pressure. The soil above the water table may become saturated through capillary action and rainfall may temporarily saturate the soil in the vadose zone. Ground water may become trapped in underground aquifers, re-emerge as surface water, or be returned to the oceans as ground water discharge.

Water stored on the surface may be retained for a long time and subject to evaporation, or detained temporarily and drained by run-off. Water may be detained in reservoirs on the surface, within the soil column, or in the deep subsurface. Water that never infiltrated the soil, or water that has re-emerged, may be stored in surface depressions and subject to evaporation, or it may run-off the surface of the land as overland flow. As a storm event proceeds storage becomes saturated and a greater proportion of water is realised as run-off. Overland flow initially traverses the surface of the land as a diffusive sheet

of water known as hill flow until it becomes concentrated as channel flow inside defined channels created by run-off driven incision (Heine et al., 2004). Channel flow is fed by surface flows, subsurface flows, and ground waters.

The rate of infiltration is highly variable due to the characteristics of the land surface and underlying soil, the saturation level of the soil, and the rate of precipitation. Horton overland flow is surface run-off generated by rainfall that is too rapid to be absorbed by the soil (Horton, 1945). Saturation overland flow occurs when soil is saturated from below by subsurface flow, usually at the bottom of hill slopes and near streams. Subsurface water in the soil flows much more slowly than Horton overland flow, therefore, only areas close to the stream contribute subsurface flows to surface channels during a storm event (Dunne et al., 1975). Surface and subsurface water flow that contributes to stream channel flow in a few days or less after a rainfall event is called storm run-off. Storm run-off may be contrasted with base flow, which is fed primarily from ground water flow and does not respond directly to individual rainfall events. The surface hydrology of a region, including the response of run-off to rainfall, is affected by topography, geology, vegetation and human activities (Chow et al., 1988).

B.2. Urban hydrology

There is a sub-discipline of hydrology called urban hydrology, which is the study of hydrological processes that are affected by urbanisation (Hall, 1984). The effect of land use change on hydrology, in particular, has become a topic of interest to researchers (DeFries and Eshleman, 2004). Urbanisation, in a general sense, is the increasing influence of urban areas within human or natural systems. Urbanisation of human settlement is driven by migration to already populated areas to form more dense communities. The social connections and opportunities for specialisation of labour enhance the economic benefit of living in urban areas, which in turn attracts more migration. Urbanisation and the agricultural practices that support it produce dramatic change to the hydrological regime of areas subject to it. The study of urban hydrology is often motivated by a desire to minimise the impact of development on the quantity and quality of surface run-off generated in urbanised areas (Wright and Leonard, 2012).

Urban catchments have many characteristics that distinguish them from rural and wilderness catchments. In the course of urbanisation, forests are felled, swamps drained, ponds and depressions regraded, and land is cleared and compacted. To house urban populations and provide places for work and recreation, towns and cities with their associated infrastructure are built. To keep dry, people build structures with impermeable roofs. To

facilitate travel, people build roads of asphalt and concrete. To feed themselves, people till soil, irrigate crops, and fertilize land. Urbanisation leads not only to more people, more streets and more buildings but also to an increased density of people, streets, and buildings.

The proliferation and concentration of impermeable surfaces typical in urban areas prevents precipitation from infiltrating the ground and causes water to collect on the surface and run-off the land. To accommodate the increase in run-off caused by urbanisation, communities typically construct a network of channels and pipes to safely transport surface water to a safe discharge point without damage to property and life. This network may also function as a foul sewer to transport wastes to be treated before discharge. Alternatively, a separate network may be created for this purpose. The urbanisation of a catchment generally increases run-off quantity and decreases run-off quality both within and downstream of the catchment being urbanised. Figure B.2 on page 284 includes built features in the hydrological cycle.

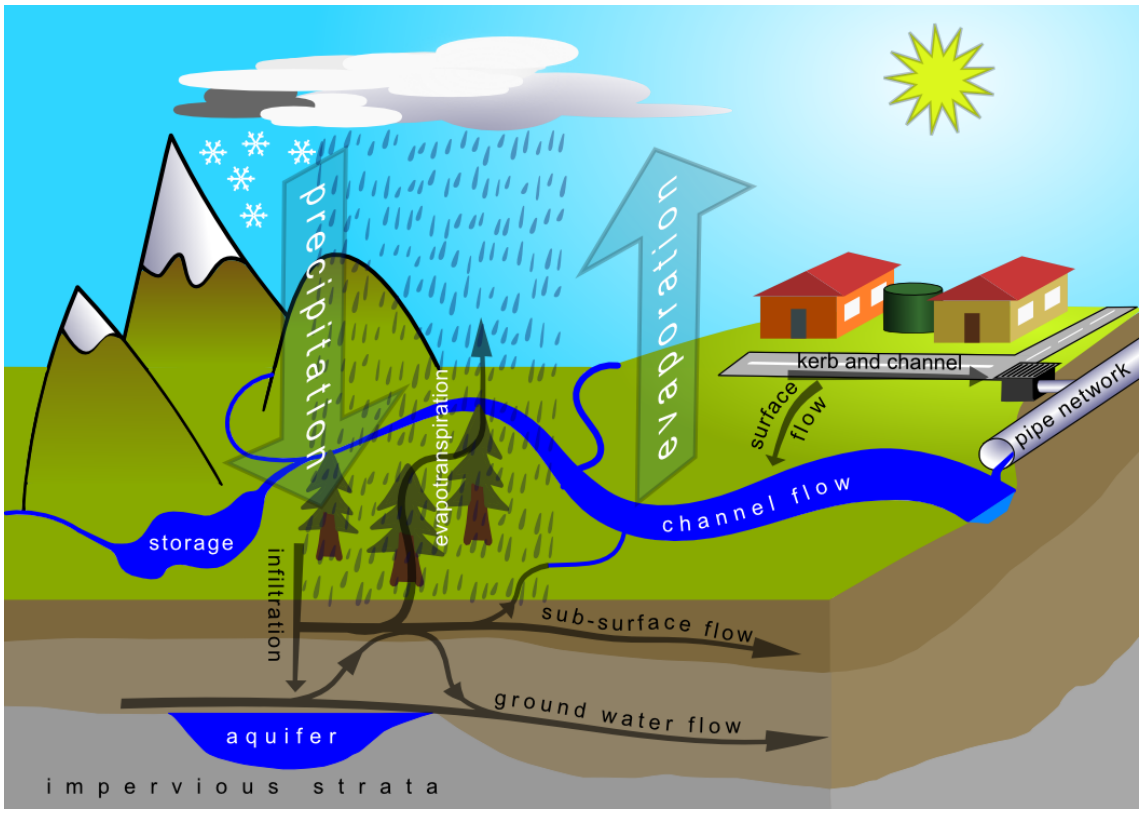


Figure B.2.: Urban hydrology.

Hall (1984) identifies three important challenges for management of urban hydrology:

1. Delivering water supply of sufficient quantity and quality to meet the needs of the population.

2. Providing flood control.
3. Controlling pollution to acceptable levels.

These issues result in the development of infrastructure to meet the demand for water, augment the natural drainage network, protect downstream areas, and dispose of wastes. Each of these three challenges are discussed below along with two further challenges.

4. Minimising erosion.
5. Mitigating climate change.

Delivering water supply of sufficient quantity and quality to meet the needs of the population

The populations of urban areas require water for drinking, cleaning, industrial, and recreational purposes. In small communities, demand for water may be met from naturally occurring sources such as rivers and springs or through small scale local infrastructure such as roof top rain water collection and wells. If a community becomes too densely populated for their water needs to be met by these means, centralised infrastructure such as dams, reservoirs and large scale pumping of ground water are developed and combined with water reticulation to deliver water to where it is needed. As cities increase in size, it becomes necessary to bring water resources from increasingly distant areas, incurring greater cost and creating competition with other centres and alternative uses such as agriculture and mining. Much of the water delivered via the water network is eventually discharged through either the stormwater or foul sewer networks.

Providing flood control

The changes urbanisation brings to the environment greatly affect the quantity of rainfall induced surface water run-off. Without mitigation, urbanisation can increase the peak discharge by a factor of five compared to an undeveloped catchment (Booth, 1991). Changes in the quantity of discharge are driven by both the increase in impermeable surfaces and the increase in velocity of water movement through a catchment. Due to the increase in impermeable surfaces, less precipitation is able to soak into the ground and a greater proportion of it is realised as surface run-off compared to an undeveloped state. Urbanisation will have greatest impact in catchments where subsurface flow once dominated (Booth, 1991).

Hard, impenetrable surfaces; storm sewers, and straighter channels all allow run-off to flow faster; which transmits water through the drainage network more quickly. Therefore,

during rainfall events, greater areas of land can simultaneously contribute flow to a single point of discharge, which leads to a greater quantity of run-off being discharged during a shorter time.

A catchment can be described as being fully developed when all parts of the catchment are contributing run-off to the outlet. According to the rational formula (Mulvaney, 1851), if water is transmitted through a catchment more quickly, the time required for a catchment to be fully realised (also known as the time of concentration) will decrease. Decreased time of concentration implies a decrease in the duration of a storm event required to fully realise the catchment. The distribution of storm intensity against storm duration indicates that shorter storms of a given intensity are more common than longer ones of the same intensity. Therefore, decreasing the time of concentration of a catchment results in a given peak discharge rate being achieved more frequently.

The increased volume of run-off generated by urbanisation is generally accommodated by the construction of engineered drainage structures such as channel networks and pipe systems. Design of drainage infrastructure and transport networks must meet the requirements of the community to protect property from flooding and allow unimpeded travel in all but the most severe and uncommon storm events. In addition, the drainage infrastructure should be resilient to demands likely to be placed on it in the future, which may include further development, changes in climate, and sea level rise.

In inland areas, drainage infrastructure that is designed solely to address the drainage problem of the immediate area may inadvertently create an external drainage problem and exacerbate flooding downstream (Hall, 1984). The external drainage problem is an issue both at the city scale and at the scale of individual neighbourhoods. One approach to address the external drainage problem is to construct artificial detention measures to perform the reservoir functions of the natural environment that have been inhibited by urbanisation. Measures such as rain gardens and detention ponds act as storage for flood waters and release their stored water slowly to reduce peak discharge.

Controlling pollution

Urban populations generate large volumes of wastewater and insanitary wastes, which may be disposed of in septic tanks or in centralised waste treatment plants that treat wastes before discharging it into rivers, lakes, and oceans. Sources of wastewater are connected to waste treatment plants by foul sewer networks. Even with a functioning foul sewer network, urbanisation frequently decreases channel water quality due to the combined effects of low flows and contamination. An increase in impermeable area reduces

ground water recharge, which reduces base flow rates between storm events. Low base flows combined with an increase in water borne pollutants leaking from the foul sewer network, and washed from streets and roofs, increases the concentration of pollutants in waterways. Quality control based on treatment plants cannot effectively address these diffuse non-point pollutants.

Hall (1984) identified land use controls and retaining vegetation as mitigation measures that hold promise to control diffuse non-point pollutants. Fletcher et al. (2013) identified a clear trend in modern hydrology management towards approaches that attempt to restore pre-development flow-regimes and water quality, to improve both the quality of the environment and the quality of life of urban residents. However, the performance of stormwater technologies for removing pollutants remains poorly quantified (Fletcher et al., 2013).

Minimising erosion

The increase in peak discharge associated with urbanisation of a catchment can lead to channel erosion and incision. Detention ponds can reduce channel erosion by reducing peak flows. However, detention ponds create extended periods of above average flow rates even whilst delivering lower peak discharge; therefore, additional erosion may still occur. Another factor that may increase stream bed erosion in urban areas is the removal of vegetation around water courses which, in addition to reducing leaf drop, shading, and temperature control also removes woody debris from the stream that would otherwise dissipate energy, thereby reducing erosion (Booth, 1991).

Mitigating climate change

Urbanisation has been shown to have measurable effects on the microclimate of the surrounding area. The effects of urbanisation on local climate include, heat islands, increased fogs, and increased precipitation (Landsberg, 1981). Urban drainage infrastructure must be resilient to these changes, in addition to the ongoing climate change and sea level rise generated by emissions of greenhouse gas. Greenhouse gas emissions are inextricably linked to urbanisation due to the dependence of urban residents on fossil fuels for transport within cities, and to transport resources into urban areas from outside the region. In addition to adaptation, careful management of urban hydrology may contribute to mitigation of both local and global climate effects.

B.3. Urban surface characteristics

Human activity shapes urban environments into forms that differ from naturally occurring areas. This section describes characteristics of urban surfaces and urban catchments, identifies challenges for hydrological modelling in urban environments, and suggests some requirements for distributed hydrological modelling in urban areas.

Urban environments are not uniform; they vary both within and between cities. This description considers urban environments typical of industrialised nations. Some elements of the urban environment are listed and categorised in Table B.1. Urban areas often have high monetary and social value. Given that the consequences of even limited flooding can be significant to affected people, it is beneficial to understand fine scale effects in high value areas.

Surfaces in the built environment can have highly variable and spatially discontinuous levels of permeability due to relatively impermeable surfaces such as concrete and asphalt being interspersed with more permeable, natural surfaces. There is also variability in the short-term storage capacity of the built environment due to the presence or absence of small surface depressions that require filling before incident rainfall can generate surface run-off. Therefore, fine spatial precision is critically important when building hydrological models of the built environment (Wright and Leonard, 2012).

Urban areas are increasingly data rich environments (Section 3.2). The surface models that are now available for urban hydrology were inconceivable a short time ago. Remotely sensed data from satellites, conventional aircraft, and Unmanned Autonomous Vehicles (UAVs) provide extensive raster imagery and dense elevation point clouds. In addition, vector databases from ground and aerial survey add to the information available. As a result of these information flows, GIS databases contain multi-dimensional spatial information about hydrologically significant components of the urban environment, such as streams and road centrelines, pipe networks, and surface level information in the form of contours or DEMs. These data are developing increasingly fine temporal and spatial resolution, due to improvements in data collection and storage technologies. Contemporary hydrological modelling techniques should maximise the benefit of these data to produce information and assists decision makers.

Urban development is an ongoing process. Not withstanding efforts by planners, engineers, and developers to develop areas whilst preserving natural flow regimes; development has numerous influences on the hydrology of urban environments and these changes can be affected more quickly than processes in unaltered environments (Fletcher et al., 2013). Therefore, hydrological modellers need to be able to adjust conveniently parts of their

models in both content and resolution to accommodate changing conditions, increased data availability, and changing significance of specific areas. A hydrologically optimal urban surface model would accommodate changes in the model without necessitating re-calculation of the whole surface, thus reducing the processing required for what if scenario modelling and the frequent changes to urban surfaces.

Table B.1.: Elements of urban landscapes.

Natural elements	Topography
	Soil types
	Plant life
	Animal influences
	Natural drainage systems
	Weather and climate
Constructions	Buildings
	Gardens
	Fences
	Retaining walls
	Sea walls
Transport networks	Roads
	Bridges
	Tunnels
Drainage networks	Spouting
	Kerb and channels
	Catch pits
	Artificial channels (canals)
	Pipe networks
	Channel flood defences
	High velocity channels
	Detention ponds
	Secondary flow paths
Water supply	Water storage for drinking water
	Subsurface water extraction

B.4. Urban hydrological modelling

Modelling techniques have been developed for modelling hydrological processes in urban areas. Fletcher et al. (2013) detail the state of the art of urban hydrological modelling. Auckland Regional Council's TP108 for hydrological modelling has since 2000 become a proxy guideline used by many Regional Councils in New Zealand (Joynes, 2009).

Fletcher et al. (2013) suggest a convergence of approaches to modelling hydrology in natural and urban areas, with spatial arrangement, network structure and sub-area be-

haviours being increasingly taken into account. Surface models from GIS databases are frequently included in hydrological models.

The high spatial variability of urban hydrology suggests either a multi-scale hierarchical raster or an irregular tessellation may be the appropriate data format for distributed hydrological models in urban areas. To be appropriate to represent urban hydrology, an urban surface model needs to be extensive, support fine and adaptive resolution, be adaptive temporally, and support efficient update.

Software

Underpinning the software used to model of hydrology are mathematical models that may be common to different applications. Sometimes software names are used as de facto names for the underlying mathematical model. However, because most models have continually evolving codes, interfaces, supporting software, parameterisation schemes, and parameter optimization packages, they should be considered a work in progress (Kampf and Burges, 2007). The following list details some of the better known hydrological models that may be applied to urban hydrological modelling. This list represents a very small sample of the many models and software packages currently in use, for further examples readers are referred to Singh and Woolhiser (2002) and Kampf and Burges (2007). Many models are in common use primarily because they are readily available, either due to support from governmental agencies or through a commercial organisation (Buytaert et al., 2008).

- TOPMODEL (Beven and Kirkby, 1979; Beven, 1995) is an early distributed model that has been very influential at a conceptual level.
- The Système Hydrologique Européen (SHE) model (Abbott et al., 1986a,b) is a catchment-scale physically based, spatially distributed model for water flow and sediment transport. The SHE model was developed by three organizations: the Institute of Hydrology (the United Kingdom), SOGREAH (France), and Danish Hydraulic Institute (Denmark). More recent derivatives include MIKE SHE (Refshaard et al., 1995), SHESED (Wicks and Bathurst, 1996) (DHI Water.Environment.Health) and SHETRAN (School of Civil Engineering and Geosciences, Newcastle University) (Ewen et al., 2000).
- EPA SWMM (Rossman, 2009) is a dynamic rainfall run-off, subsurface run-off simulation model used for single-event to long-term (continuous) simulation of the surface/subsurface hydrology quantity and quality from primarily urban/suburban areas. SWMM includes 1-D pipe flow. SWMM is a derivative of the earlier Stanford Watershed Model (SWM). EPA SWMM is public domain software that may be

freely copied and distributed. It is implemented widely in other software, including the commercial packages XP-SWMM and InfoSWMM.

- The Hydrologic Engineering Center (HEC) within the U.S. Army Corps of Engineers is responsible for HEC-RAS (Tate et al., 2002) and HEC-HMS, which have lumped representations of the model subunits, but the subunits are connected to each other to enable 1-D routing of water through 2-D space. It is designed to be applicable in a wide range of geographic areas and hydrological problems including water supply and flood hydrology in large river basins, and small urban or natural catchment run-off. Hydrographs produced by the program are used directly or in conjunction with other software for studies of water availability, urban drainage, flow forecasting, future urbanization impact, reservoir spillway design, flood damage reduction, floodplain regulation, and systems operation.
- XP-UDD/XP-SWMM, TUFLOW, and INFOWORKS ICM are contemporary commercial hydrological modelling packages with large user bases.
- State of the art academic models that represent governing conservation equations in 3-D for subsurface processes coupled with 2-D surface flows using diffusion wave approximation of the de St. Venant equations include MODHMS (Panday and Huyakorn, 2004) and WASH123D (Yeh et al., 1998).

C. HIP Calculation examples

C.1. HIP Location

RHSM-hex HIP^7 to xy

Let us take, for example, a HIP^7 dataset with cellsize (cs) = 1m and calculate the New Zealand Transverse Mercator 2000 (NZTM) coordinates of the centre of cell 24043, given that the centre of cell 00000 = (1406262.00N, 4916908.00E) in NZTM.

HIP_i is the digit at index i, i.e. $HIP_4,HIP_3,HIP_2,HIP_1,HIP_0 = 2,4,0,4,3$. From Table 6.1a, $\mathbf{B} = \begin{bmatrix} 2 & -\sqrt{3} \\ \sqrt{3} & 2 \end{bmatrix}$ and $\mathbf{C} = \begin{bmatrix} cs \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

for $i = 0$: HIP_0= 3, $\mathbf{A}_3$ can be found in Table 6.1a.

$$\mathbf{A}_3\mathbf{B}^0\mathbf{C} = \begin{bmatrix} -1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix} \begin{bmatrix} 2 & -\sqrt{3} \\ \sqrt{3} & 2 \end{bmatrix}^0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1/2 \\ \sqrt{3}/2 \end{bmatrix} \approx \begin{bmatrix} -0.500 \\ 0.866 \end{bmatrix}$$

for $i = 1$: HIP_1= 4.

$$\mathbf{A}_4\mathbf{B}^1\mathbf{C} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 2 & -\sqrt{3} \\ \sqrt{3} & 2 \end{bmatrix}^1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -2 \\ -\sqrt{3} \end{bmatrix} \approx \begin{bmatrix} -2.000 \\ -1.732 \end{bmatrix}$$

for $i = 2$: HIP_2= 0.

$$\mathbf{A}_0\mathbf{B}^2\mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & -\sqrt{3} \\ \sqrt{3} & 2 \end{bmatrix}^2 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.000 \\ 0.000 \end{bmatrix}$$

for $i = 3$: HIP_3= 4.

$$\mathbf{A}_4\mathbf{B}^3\mathbf{C} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 2 & -\sqrt{3} \\ \sqrt{3} & 2 \end{bmatrix}^3 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ -9\sqrt{3} \end{bmatrix} \approx \begin{bmatrix} 10.000 \\ -15.588 \end{bmatrix}$$

for $i = 4$: HIP_4= 2.

$$\mathbf{A}_2\mathbf{B}^4\mathbf{C} = \begin{bmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{bmatrix} \begin{bmatrix} 2 & -\sqrt{3} \\ \sqrt{3} & 2 \end{bmatrix}^4 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -71/2 \\ -39/2\sqrt{3} \end{bmatrix} \approx \begin{bmatrix} -35.500 \\ -33.775 \end{bmatrix}$$

$$\begin{bmatrix} x_\triangle \\ y_\triangle \end{bmatrix} = \sum_{i=0}^{\lambda-1} \mathbf{A}_d \mathbf{B}^i \mathbf{C} \approx \begin{bmatrix} -0.500 \\ 0.866 \end{bmatrix} + \begin{bmatrix} -2.000 \\ -1.732 \end{bmatrix} + \begin{bmatrix} 0.000 \\ 0.000 \end{bmatrix} + \begin{bmatrix} 10.000 \\ -15.588 \end{bmatrix} + \begin{bmatrix} -35.500 \\ -33.775 \end{bmatrix} = \begin{bmatrix} -28.000 \\ -50.229 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_\triangle \\ y_\triangle \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \approx \begin{bmatrix} -28.000 \\ -50.229 \end{bmatrix} + \begin{bmatrix} 4916908.00 \\ 1406262.00 \end{bmatrix} = \begin{bmatrix} 4916880.00 \\ 1406211.77 \end{bmatrix}$$

RHSM-tri HIP^4 to xy

In RHSM-tri, the alternating orientation is affected by alternating the sign of $\mathbf{C}$ for every coarser occurrence of HIP digit 0 in the HIP^4 ordinate. Let us repeat a similar exercise for a $4\lambda HIP^4$ dataset with cellsize $(cs) = 1$m and calculate the vector $\begin{bmatrix} x_\triangle \\ y_\triangle \end{bmatrix}$ between the origin at the centre of cell 0000 and centre of cell 2103.

HIP_i is the digit at index i, i.e. $HIP_{3,}HIP_{2,}HIP_{1,}HIP_0 = 2, 1, 0, 3$ and $\lambda = 4$. From Table 6.1a, $\mathbf{B} = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}$ and $\mathbf{C} = \begin{cases} \text{if } \left(\lambda - \sum_{\text{HIP}_\text{i}}^{HIP_\lambda} 1\left[HIP_\text{i} = 0\right]\right) \text{is even} & \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \text{if } \left(\lambda - \sum_{\text{HIP}_\text{i}}^{\text{HIP}_\lambda} 1\left[HIP_\text{i} = 0\right]\right) \text{is odd} & \begin{bmatrix} -1 \\ 0 \end{bmatrix} \end{cases}$

for $i = 0$: $HIP_0 = 3$, $\mathbf{A}_3$ can be found in Table 6.1a. There is one instance of 0 in $HIP_{3,}HIP_{2,}HIP_{1,}HIP_0 \Rightarrow 4 - 1 = 3$ is odd $\therefore \mathbf{C} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$$\mathbf{A}_3\mathbf{B}^0\mathbf{C} = \begin{bmatrix} \sqrt{3}/2 & 1/2 \\ -1/2 & \sqrt{3}/2 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}^0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \sqrt{3}/2 \\ -1/2 \end{bmatrix} \approx \begin{bmatrix} 0.866 \\ -0.500 \end{bmatrix}$$

for $i = 1$: $HIP_1 = 0$. There is one instance of 0 in $HIP_{3,}HIP_{2,}HIP_1 \Rightarrow 4 - 1 = 3$ is odd $\therefore \mathbf{C} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$$\mathbf{A}_0\mathbf{B}^1\mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}^1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.000 \\ 0.000 \end{bmatrix}$$

for $i = 2$: $HIP_2 = 1$. There are no instances of 0 in $HIP_{3,}HIP_2 \Rightarrow 4 - 0 = 4$ is even $\therefore \mathbf{C} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$

$$\mathbf{A}_1\mathbf{B}^2\mathbf{C} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}^2\begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.000 \\ -4.000 \end{bmatrix}$$

for $i = 3$: $HIP_3 = 2$. $HIP_3 \neq 0 \Rightarrow 4 - 0 = 4$ is even $\therefore \mathbf{C} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$

$$\mathbf{A}_2\mathbf{B}^3\mathbf{C} = \begin{bmatrix} -\sqrt{3}/2 & 1/2 \\ -1/2 & -\sqrt{3}/2 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}^3\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 4\sqrt{3} \\ 4 \end{bmatrix} \approx \begin{bmatrix} 6.928 \\ 4.000 \end{bmatrix}$$

$$\begin{bmatrix} x_\triangle \\ y_\triangle \end{bmatrix} = \sum_{i=0}^{\lambda-1}\mathbf{A}_d\mathbf{B}^i\mathbf{C} \approx \begin{bmatrix} 0.866 \\ -0.500 \end{bmatrix} + \begin{bmatrix} 0.000 \\ 0.000 \end{bmatrix} + \begin{bmatrix} 0.000 \\ -4.000 \end{bmatrix} + \begin{bmatrix} 6.928 \\ 4.000 \end{bmatrix} = \begin{bmatrix} 7.794 \\ -0.500 \end{bmatrix}$$

C.2. HIP Arithmetic

Examples of HIP arithmetic are given below.

Addition

As an example, $HIP^9\,062 \oplus 024$ is determined below. From Table 6.2b: $2 \oplus 4 = 37$, $3 \oplus 6 = 5$, $5 \oplus 2 = 3$.

$$\begin{array}{cccc} & & {\scriptstyle 3} & \\ & 0 & 6 & 2 \\ \oplus & 0 & 2 & 4 \\ \hline & 0 & 3 & 7 \end{array}$$

Negation

As an example, $HIP^7\,062 \ominus 024$ is determined below.

First the negation of 024 is found by substituting the digits using Table 6.1c.

$\ominus 024 = 051$

$062 \ominus 024 = 062 \oplus 051 = 525$

Neighbourhood

The neighbourhood of $HIP^7\,024$ is calculated below.

$$\begin{aligned} neighbourhood[024] &= [024 \oplus 001, 024 \oplus 002, 024 \oplus 003, 024 \oplus 004, 024 \oplus 005, 024 \oplus 006] \\ &= [020, 023, 261, 266, 032, 025] \end{aligned}$$

Bibliography

Abbott, M., J. Bathurst, J. Cunge, P. O'Connell, and J. Rasmussen: 1986a, 'An introduction to the European Hydrological System - Systeme Hydrologique Europeen, "SHE", 1: History and philosophy of a physically-based, distributed modelling system'. *Journal of Hydrology* **87**(1), 45 – 59.

Abbott, M., J. Bathurst, J. Cunge, P. O'Connell, and J. Rasmussen: 1986b, 'An introduction to the European Hydrological System - Systeme Hydrologique Europeen, "SHE", 2: Structure of a physically-based, distributed modelling system'. *Journal of Hydrology* **87**(1), 61 – 77.

Abdelguerfi, M., C. Wynne, E. Cooper, and L. Roy: 1998, 'Representation of 3-D elevation in terrain databases using hierarchical triangulated irregular networks: a comparative analysis'. *International Journal of Geographical Information Science* **12**(8), 853–873.

Agarwal, P., L. Arge, and A. Danner: 2006, 'From Point Cloud to Grid DEM: A Scalable Approach'. *Progress in Spatial Data Handling* **13**, 771–788.

Agüero, J. C., A. Feuer, and G. C. Goodwin: 2003, 'Terrain modelling via triangular regular networks'. In: *MODSIM 2003*, Vol. XXXIII. Townsville, Australia.

Ai, T. and J. Li: 2009, 'A DEM generalization by minor valley branch detection and grid filling'. **65**, 198–207.

Alted, F., I. Vilata, et al.: 2002, 'PyTables: Hierarchical Datasets in Python'. http://www.pytables.org/ Accessed December 2016.

Anderberg, S.: 2005, 'Systems Analysis in Geography'. In: M.-O. Olsson and G. Sjöstedt (eds.): *Systems Approaches and Their Application*. Springer Netherlands, pp. 79–93.

Anderson, E. S., J. A. Thompson, D. A. Crouse, and R. E. Austin: 2006, 'Horizontal resolution and data density effects on remotely sensed LIDAR-based DEM'. *Geoderma* **132**(3-4), 406–415.

Anderson, M. G. and P. D. Bates (eds.): 2001, *Model Validation: Perspectives in Hydrological Science*. london: John Wiley and Sons.

Andersson, B., R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo, and B. Van Wachem: 2011, *Computational fluid dynamics for engineers*. Cambridge University Press.

Arnold, N.: 2010, 'A new approach for dealing with depressions in digital elevation models when calculating flow accumulation values'. *Progress in Physical Geography* **34**(6), 781–809.

Atkinson, P. M. and N. J. Tate: 2000, 'Spatial Scale Problems and Geostatistical Solutions: A Review'. *The Professional Geographer* **52**(4), 607–623.

Band, L. E.: 1986, 'Topographic Partition of Watersheds with Digital Elevation Models'. *Water Resources Research* **22**(1), 15–24.

Bandaragoda, C.: 2008, 'Distributed Hydrologic Modeling For Streamflow Prediction At Ungauged Basins'. Ph.D. thesis, Utah State University.

Barnsley, M. J.: 2007, *Environmental modeling: a practical introduction*. Boca Raton: CRC.

Bell, S. B. M., B. M. Diaz, and F. C. Holroyd: 1989, 'The HoR quadtree: An optimal structure based on a non-square 4-shape'. In: *Proc. Conf. on Mathematics in Remote Sensing*. Oxford, Clarendon Press.

Beven, K.: 1995, 'Linking parameters across scales: Subgrid parameterizations and scale dependent hydrological models'. *Hydrological Processes* **9**(5-6), 507–525.

Beven, K.: 1996, 'Equifinality and Uncertainty in Geomorphological Modelling'. In: *The scientific nature of geomorphology: proceedings of the 27th Binghamton Symposium in Geomorphology*, Vol. 27. John Wiley & Sons, pp. 289–313.

Beven, K. J.: 2001, 'Dalton Medal Lecture: How far can we go in distributed hydrological modelling?'. *Hydrology and Earth System Sciences* **5**(1), 1–12.

Beven, K. J., J. Freer, and B. H. K. Schulz: 2000, 'The use of generalised likelihood measures for uncertainty estimation in high order models of environmental systems'. In: W. J. Fitzgerald, R. Smith, A. T. Walden, and P. C. Young (eds.): *Nonlinear and nonstationary signal processing*. Cambridge University Press, pp. 115–151.

Beven, K. J. and M. J. Kirkby: 1979, 'A physically based, variable contributing area model of basin hydrology'. *Hydrological Sciences-Bulletin-des Sciences Hydrologiques* **24**(1), 43–69.

Biggs, N. L., E. K. Lloyd, and R. J. Wilson: 1998, *Graph Theory 1736-1936*. Oxford: Clarendon Press.

Bjørke, J. T. and S. Nilsen: 2003, 'Wavelets applied to simplification of digital terrain models'. *International Journal of Geographical Information Science* **17**(7), 601–621.

Blöschl, g. and m. Sivapalan: 1995, 'Scale issues in hydrological modelling: A review'. *Hydrol. Process.* **9**(3-4), 251–290.

BMT WBM: 2007-2015, 'TUFLOW Flood and Coastal Simulation Software'. http://www.tuflow.com/Default.aspx Accessed 4/12/2014.

Bolles, R. C., H. H. Baker, and D. H. Marimont: 1987, 'Epipolar-plane image analysis: An approach to determining structure from motion'. *International Journal of Computer Vision* **1**(1), 7–55.

Booth, D. B.: 1991, 'Urbanization and the natural drainage system - Impactacts, solutions, and prognoses'. *The Northwestern Environmental Journal* **7**(1), 93–118.

Boots, B. and N. Shiode: 2003, 'Recursive Voronoi diagrams'. *Environment and Planning B: Planning and Design* **30**(1), 113–124.

Borna, K., A. Moore, and P. Sirguey: 2014, 'Towards a vector agent modelling approach for remote sensing image classification'. *Journal of Spatial Science* **59**(2), 283–296.

Braun, J. and M. Sambridge: 1997, 'Modelling landscape evolution on geological time scales: a new method based on irregular spatial discretization'. *Basin Research* **9**(1), 27–52.

Brimkov, V. E. and R. P. Barneva: 2001, '"Honeycomb" vs Square and Cubic Models'. *Electronic Notes in Theoretical Computer Science* **46**, 321–338.

Burrough, P. A., P. F. M. van Gaans, and R. A. MacMillan: 2000, 'High-resolution landform classification using fuzzy k-means'. *Fuzzy Sets and Systems* **113**(1), 37–52.

Buytaert, W., D. Reusser, S. Krause, and J.-P. Renaud: 2008, 'Why can't we do better than Topmodel?'. *Hydrological Processes* **22**(20), 4175–4179.

Callow, J. N., K. P. Van Niel, and G. S. Boggs: 2007, 'How does modifying a DEM to reflect known hydrology affect subsequent terrain analysis?'. *Journal of Hydrology* **332**(1-2), 30–39.

Catchment Simulation Solutions: 2008, 'CatchmentSIM Automated Catchment Mapping for 12d Model'. Brochure. Online at http://csse.com.au/images/documents/12d_NZ.pdf (last accessed 2018).

Chaplot, V., F. Darboux, H. Bourennane, S. Leguédois, N. Silvera, and K. Phachomphon: 2006, 'Accuracy of interpolation techniques for the derivation of digital elevation models in relation to landform types and data density'. *Geomorphology* **77**(1-2), 126–141.

Chen, A. S., B. Evans, S. Djordjević, and D. A. Savić: 2012, 'Multi-layered coarse grid modelling in 2D urban flood simulations'. *Journal of Hydrology* **470-471**, 1–11.

Cherkauer, K. A., L. C. Bowling, and D. P. Lettenmaier: 2003, 'Variable infiltration capacity cold land process model updates'. *Global and Planetary Change* **38**(12), 151–159. Project for Intercomparison of Land-surface Parameterization Schemes, Phase 2(e).

Chorowicz, J., C. Ichoku, S. Raizanoff, K. Youn-Jong, and B. Cervelle: 1992, 'A combined algorithm for automated drainage network extraction'. *Water Resources Research* **28**(5), 1293–1302.

Chou, T. Y., W. T. Lin, C. Y. Lin, W. C. Chou, and P. H. Huang: 2004, 'Application of the PROMETHEE technique to determine depression outlet location and flow direction in DEM'. *Journal of Hydrology* **287**(1-4), 49–61.

Chow, V. T., D. R. Maidment, and L. W. Mays: 1988, *Applied Hydrology*, McGraw-Hill series in water resources and environmental engineering. McGraw-Hill.

Cieplak, M., A. Giacometti, A. Maritan, A. Rinaldo, I. Rodriguez-Iturbe, and J. R. Banavar: 1998, 'Models of fractal river basins'. *Journal of Statistical Physics* **91**(1-2), 1–15.

Cignoni, P., E. Puppo, and R. Scopigno: 1997, 'Representation and visualization of terrain surfaces at variable resolution'. *The Visual Computer* **13**(5), 199–217.

Clark, M. P. and D. Kavetski: 2010, 'Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes'. *Water Resour. Res.* **46**(10), W10510.

Clarke, R.: 1973, 'A review of some mathematical models used in hydrology, with observations on their calibration and use'. *Journal of Hydrology* **19**(1), 1 – 20.

Comić, L., L. De Floriani, and L. Papaleo: 2005, 'Morse-Smale Decompositions for Modeling Terrain Knowledge'. In: A. G. Cohn and D. M. Mark (eds.): *Lecture Notes in Computer Science*, Vol. 3693. Springer Berlin / Heidelberg, pp. 426–444.

Condat, L. and D. Van De Ville: 2008, 'New optimized spline functions for interpolation on the hexagonal lattice'. In: *15th International Conference on Image Processing*. pp. 1256–1260.

Condat, L., D. V. D. Ville, and B. Forster-Heinlein: 2008, 'Reversible, fast, and high-quality grid conversions'. *IEEE Transactions on Image Processing* **17**(5), 679–693.

Costa-Cabral, M. C. and S. J. Burges: 1994, 'Digital elevation model networks (DEMON): A model of flow over hillslopes for computation of contributing and dispersal areas'. *Water Resources Research* **30**(6), 1681–1692.

Couprie, M., L. Najman, and G. Bertrand: 2005, 'Quasi-linear algorithms for the topological watershed'. *Journal of Mathematical Imaging and Vision* **22**(2-3), 231–249.

Cressie, N.: 1991, *Statistics for Spatial Data*. John Wiley & Sons.

Dakowicz, M. and C. M. Gold: 2007, 'Finite difference method runoff modelling using voronoi cells'. In: *Proceedings of the 5th ISPRS Workshop on Dynamic and Multi-dimensional GIS*. Urumchi, China, pp. 55–60.

Dalyot, S., E. Keinan, and Y. b. Doytsher: 2008, 'Toward continuous and updated 3D geospatial terrain modelling based on DTM and LiDAR data'. In: *Proceedings of the ISPRS*. Beijing, China.

Danovaro, E., L. De Floriani, P. Magillo, M. M. Mesmoudi, and E. Puppo: 2003, 'Morphology-driven simplification and multiresolution modeling of terrains'. In: *Proceedings of the 11th ACM international symposium on Advances in geographic information systems*. New York, NY, USA, pp. 63–70, ACM.

Danovaro, E., L. De Floriani, P. Magillo, E. Puppo, and D. Sobrero: 2006, 'Level-of-detail for data analysis and exploration: A historical overview and some new perspectives'. *Computers & Graphics* **30**(3), 334–344.

de Floriani, L.: 1989, 'A Pyramidal Data Structure for Triangle-Based Surface Description'. *IEEE Computer Graphics and Applications* **9**, 67–78.

de Floriani, L., L. Kobbelt, and E. Puppo: 2005, 'A Survey on Data Structures for Level-of-Detail Models'. In: N. Dodgson, M. Floater, and M. Sabin (eds.): *Mathematics and Visualization*. Springer Berlin Heidelberg, pp. 49–74.

de Sousa, L., F. Nery, R. Sousa, and J. Matos: 2006, 'Assessing the accuracy of hexagonal versus square tiled grids in preserving DEM surface flow directions'. In: M. Caetano and M. Painho (eds.): *the 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences (Accuracy 2006)*. Lisbon.

de St. Venant, B.: 1871, 'Theorie du mouvement non-permanent des eaux, avec application aux crues des riveres et a l'introduction des Marees dans leur lit'. *Comptes rendus des seances de l'Academie des Sciences* **73**, 148–154.

DeFries, R. and K. N. Eshleman: 2004, 'Land-use change and hydrologic processes: a major focus for the future'. *Hydrological Processes* **18**(11), 2183–2186.

Demaret, L., N. Dyn, M. Floater, and A. Iske: 2005, 'Adaptive Thinning for Terrain Modelling and Image Compression'. In: N. A. Dodgson, M. S. Floater, and M. A. Sabin (eds.): *Mathematics and Visualization*. Springer Berlin Heidelberg, pp. 319–338.

Deng, Y.: 2007, 'New trends in digital terrain analysis: landform definition, representation, and classification.'. *Progress in Physical Geography* **31**(4), 405–419.

Dinesh, S.: 2008, 'Extraction of Hydrological Features from Digital Elevation Models Using Morphological Thinning'. *Asian journal of scientific research* **1**(4), 310.

Dobrowolski, T.: 2007, 'Generating Fractal Tiles using Voronoi Diagrams'. In: *ISVD 2007 : 4th International Symposium on Voronoi Diagrams in Science and Engineering 2007*. pp. 251–254.

Döll, P. and B. Lehner: 2002, 'Validation of a new global 30-min drainage direction map'. *Journal of Hydrology* **258**(1-4), 214–231.

Dooge, J. C. I.: 1986, 'Looking for hydrologic laws'. *Water Resources Research* **22**(9S), 46S–58S.

Dowling, T., A. M. Read, and J. C. Gallant: 2009, 'Very high resolution DEM acquisition at low cost using a digital camera and free software'. In: *18th World IMACS / MODSIM Congress,*. Cairns, Australia.

Doytsher, Y., S. Dalyot, and Y. Katzil: 2009, 'Digital Terrain Models: A Tool for Establishing Reliable and Qualitative Environmental Control Processes'. In: R. D. Amicis, R. Stojanovic, and G. Conti (eds.): *NATO Science for Peace and Security Series*. Springer Netherlands, pp. 215–234.

Dřaguţ, L., C. Eisank, and T. Strasser: 2011, 'Local variance for multi-scale analysis in geomorphometry'. *Geomorphology* **130**(3-4), 162–172.

Duan, Q., S. Sorooshian, and V. Gupta: 1992, 'Effective and efficient global optimization for conceptual rainfall-runoff models'. *Water Resources Research* **28**(4), 1015–1031.

Duke, G. D., S. W. Kienzle, D. L. Johnson, and J. M. Byrne: 2006, 'Incorporating ancillary data to refine anthropogenically modified overland flow paths'. *Hydrological Processes* **20**(8), 1827–1843.

Dungan, J. L.: 2001, 'Scaling up and scaling down: the relevance of the support effect on remote sensing of vegetation'. In: N. Tate and P. M. Atkinson (eds.): *Modelling scale in geographical information science*. John Wiley & Sons, pp. 221–235.

Dunne, T., T. Moore, and C. H. Taylor: 1975, 'Recognition and prediction of runoff-producing zones in humid regions'. *Hydrological Sciences-Bulletin-des Sciences Hydrologiques* **20.3**, 305–327.

Endreny, T. A. and E. F. Wood: 2001, 'Representing elevation uncertainty in runoff modelling and flowpath mapping'. *Hydrological Processes* **15**(12), 2223–2236.

Erdogan, S.: 2009, 'Modelling the spatial distribution of DEM error with geographically weighted regression: An experimental study'. *Computers & Geosciences* **36**(1), 34–43.

Erskine, R. H., T. R. Green, J. A. Ramirez, and L. H. MacDonald: 2006, 'Comparison of grid-based algorithms for computing upslope contributing area'. *Water Resour. Res.* **42**(9), W09416.

Evans, I.: 2003, 'Scale-specific landforms and aspects of the land surface'. In: I. Evans, R. Dikau, E. Tokunaga, H. Ohmori, and M. Hirano (eds.): *Concepts and Modelling in Geomorphology: International Perspectives*. Tokyo: Terrapub, pp. 61–86.

Evans, J. S. and A. T. Hudak: 2007, 'A multiscale curvature algorithm for classifying discrete return LiDAR in forested environments'. *IEEE Magazine Transactions on Geoscience and Remote Sensing* **45**(4), 1029–1038.

Ewen, J., G. Parkin, and P. O'Connell: 2000, 'SHETRAN: Distributed River Basin Flow and Transport Modeling System'. *Journal of Hydrologic Engineering* **5**(3), 250–258.

Faille, F. and M. Petrou: 2010, 'Invariant image reconstruction from irregular samples and hexagonal grid splines'. *Image and Vision Computing* **28**(8), 1173–1183.

Fairfield, J. and P. Leymarie: 1991, 'Drainage Networks from Grid Digital Elevation Models'. *Water Resources Research* **27**(5), 709–717.

Famiglietti, J. S. and E. F. Wood: 1994, 'Multiscale modeling of spatially variable water and energy balance processes'. *Water Resources Research* **30**(11), 3061–3078.

Fedorov, E.: 1891, 'Simmetrija na ploskosti'. *Zapiski Imperatorskogo Sant-Petersburgskogo Mineralogicheskogo Obshchestva* **2**(28), 245–291. In russian.

Feick, R. and B. Boots: 2005, 'Variable Resolution Spatial Interpolation Using the Simple Recursive Point Voronoi Diagram'. *Geographical Analysis* **37**(2), 225–243.

Fisher, P.: 1998, 'Improved modeling of elevation error with Geostatistics'. *GeoInformatica* **2**(3), 215–233.

Fleming, M. D. and R. M. Hoffer: 1979, 'Machine processing of Landsat MSS data and DMA topographic data for forest cover type mapping'. Technical Report 80, Laboratory for Applications of Remote Sensing.

Fletcher, T., H. Andrieu, and P. Hamel: 2013, 'Understanding, management and modelling of urban hydrology and its consequences for receiving waters: A state of the art'. *Advances in Water Resources* **51**(0), 261–279.

Florinsky, I. V. and G. A. Kuryakova: 2000, 'Determination of grid size for digital terrain modelling in landscape investigations - exemplifed by soil moisture distribution at a micro-scale'. *International Journal of Geographical Information Science* **14**(8), 815–832.

Fortune, S.: 1987, 'A sweepline algorithm for Voronoi diagrams'. *Algorithmica* **2**(1), 153–174.

Francos, A., F. J. Elorza, F. Bouraoui, G. Bidoglio, and L. Galbiati: 2003, 'Sensitivity analysis of distributed environmental simulation models: understanding the model be-

haviour in hydrological studies at the catchment scale'. *Reliability Engineering & System Safety* **79**(2), 205–218.

Franklin, S. E.: 1987, 'Geomorphometric processing of digital elevation models'. *Computers & Geosciences* **13**(6), 603–609.

Franklin, W. R.: 2010, 'Towards a mathematics of terrain'. In: *20th Annual Fall Workshop on Computational Geometry*.

Fread, D. L.: 1993, 'Flow routing'. In: D. R. Maidment (ed.): *Handbook of Hydrology*. New York: McGraw-Hil, Chapt. 10.11-10.36.

Freeman, T. G.: 1991, 'Calculating catchment area with divergent flow based on a regular grid'. *Computers and Geosciences* **17**(3), 413–422.

Freeze, R. and R. Harlan: 1969, 'Blueprint for a physically-based, digitally-simulated hydrologic response model'. *Journal of Hydrology* **9**(3), 237 – 258.

Freeze, R. A.: 1974, 'Streamflow generation'. *Reviews of Geophysics* **12**(4), 627–647.

Gaitan, S., M.-c. ten Veldhuis, and N. van de Giesen: 2015, 'Spatial Distribution of Flood Incidents Along Urban Overland Flow-Paths'. *Water Resources Management* **29**(9), 3387–3399.

Gallant, J. C. and M. F. Hutchinson: 1997, 'Scale dependence in terrain analysis'. *Mathematics and Computers in Simulation* **43**(3-6), 313–321.

Gallant, J. C. and M. F. Hutchinson: 2011, 'A differential equation for specific catchment area'. *Water Resour. Res.* **47**(5), W05535.

Gandolfi, C. and G. B. Bischetti: 1997, 'Influence of the drainage network identification method on geomorphological properties and hydrological response'. *Hydrological Processes* **11**(4), 353–375.

Garbrecht, J. and L. Martz: 1994, 'Grid size dependency of parameters extracted from digital elevation models'. *Computers and Geosciences* **20**(1), 85–87.

Garbrecht, J. and L. W. Martz: 1993, 'Network and subwatershed parameters extracted from digital elevation models: the Bills Creek experience'. *Journal of the American Water Resources Association* **29**(6), 909–916.

Garbrecht, J. and L. W. Martz: 1997, 'The assignment of drainage direction over flat surfaces in raster digital elevation models'. *Journal of Hydrology* **193**(1-4), 204–213.

Garbrecht, J. and L. W. Martz: 2000, 'Digital elevation model issues in water resources modeling'. In: D. R. Maidment and D. Djokic (eds.): *Hydrologic and Hydraulic Modeling Support with Geographical Information Systems*. Redlands, CA: ESRI Press, pp. 1 – 28.

Gargantini, I.: 1982, 'An Effective Way to Represent Quadtrees'. *Commun. ACM* **25**(12), 905–910.

Gerstner, T.: 1999, 'Adaptive hierarchical methods for landscape representation and analysis'. In: S. Hergarten and H. Neugebauer (eds.): *Lecture Notes in Earth Sciences*, Vol. 78. Springer Berlin / Heidelberg, pp. 75–92.

Gerstner, T.: 2003, 'Multiresolution Compression and Visualization of Global Topographic Data'. *Geoinformatica* **7**(1), 7–32.

Getirana, A. C. V., M. P. Bonnet, and J. M. Martinez: 2009, 'Evaluating parameter effects in a DEM "burning" process based on land cover data'. *Hydrological Processes* **23**(16), 2316–2325.

Gibson, L. and D. Lucas: 1982, 'Spatial data processing using generalized balanced ternary'. In: *Proceedings of the IEEE Conference on Pattern Recognition and Image Processing*.

Goepfert, J. and F. Rottensteiner: 2009, 'Adaptation of roads to ALS data by means of network snakes'. *IntArchPhRS* **38**(3/W8), 24–29.

Golay, M. J. E.: 1969, 'Hexagonal Parallel Pattern Transformations'. *Ieee Transactions on Computers* **C 18**(8), 733.

Gold, C.: 2009, 'A Common Spatial Model for GIS'. In: *Lecture Notes in Geoinformation and Cartography*. Springer Berlin Heidelberg, pp. 79–94.

Gonçalves, G. and J. Santos: 2005, 'Propagation of dem uncertainty: an intervalarithmetic approach'. In: *XXII International Cartographic Conference*.

Gong, J. and J. Xie: 2009, 'Extraction of drainage networks from large terrain datasets using high throughput computing'. *Computers & Geosciences* **35**(2), 337–346.

Goodchild, M. F.: 2001, 'Models of scale and scales of modelling'. In: N. Tate and P. M. Atkinson (eds.): *Modelling scale in geographical information science*. John Wiley & Sons, pp. 3–10.

Goodchild, M. F.: 2011, 'Scale in GIS: An overview'. *Geomorphology* **130**(12), 5 – 9. Scale Issues in Geomorphology.

Goodchild, M. F. and D. M. Mark: 1987, 'The Fractal Nature of Geographic Phenomena'. *Annals of the Association of American Geographers* **77**(2), 265–278.

Goodchild, M. F. and Y. Shiren: 1992, 'A hierarchical spatial data structure for global geographic information systems'. *CVGIP: Graphical Models and Image Processing* **54**(1), 31 – 44.

Goodchild, M. F., M. Yuan, and T. J. Cova: 2007, 'Towards a general theory of geographic representation in GIS'. *International Journal of Geographical Information Science* **21**(3), 239–260.

Grimaldi, S., F. Nardi, F. D. Benedetto, E. Istanbulluoglu, and R. L. Bras: 2007, 'A physically-based method for removing pits in digital elevation models'. *Advances in Water Resources* **30**(10), 2151–2158.

Grimaldi, S., V. Teles, and R. L. Bras: 2004, 'Sensitivity of a physically based method for terrain interpolation to initial conditions and its conditioning on stream location'. *Earth Surface Processes and Landforms* **29**(5), 587–597.

Grimaldi, S., V. Teles, and R. L. Bras: 2005, 'Preserving first and second moments of the slope area relationship during the interpolation of digital elevation models'. *Advances in Water Resources* **28**(6), 583–588.

Gruber, S. and S. Peckham: 2009, 'Chapter 7 Land-Surface Parameters and Objects in Hydrology'. In: H. Tomislav and I. R. Hannes (eds.): *Developments in Soil Science*, Vol. Volume 33. Elsevier, pp. 171–194.

Grunbaum, B. and G. C. Shephard: 1977, 'Tilings by Regular Polygons'. *Mathematics Magazine* **50**(5), 227–247.

Guibas, L. and J. Stolfi: 1985, 'Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi'. *ACM Trans. Graph.* **4**(2), 74–123.

Gupta, H. V., S. Sorooshian, and P. O. Yapo: 1998, 'Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information'. *Water Resources Research* **34**(4), 751–763.

Gyasi-Agyei, Y., G. Willgoose, and F. P. D. Troch: 1995, 'Effects of vertical resolution and map scale of digital elevation models on geomorphological parameters used in hydrology'. *Hydrological Processes* **9**(3-4), 363–382.

Hagen, A.: 2003, 'Fuzzy set approach to assessing similarity of categorical maps'. *International Journal of Geographical Information Science* **17**(3), 235–249.

Hall, M. J.: 1984, *Urban Hydrology*. London and New York: Elsevier Applied Science Publishers.

Hannah, J.: 2009, 'Heights, height Systems and height Transfer'. University of Otago SURV302 Course Lecture Notes.

Harrison, J. M. and C.-P. Lo: 1996, 'PC-based two-dimensional discrete fourier transform programs for terrain analysis'. *Computers & Geosciences* **22**(4), 419–424.

Heckbert, P. S. and M. Garland: 1997, 'Survey of polygonal surface simplification algorithms'. Technical report, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Heine, R. A., C. L. Lant, and R. R. Sengupta: 2004, 'Development and comparison of approaches for automated mapping of stream channel networks'. *Annals of the Association of American Geographers* **94**(3), 477–490.

Heiskanen, W. and H. Moritz: 1967, 'Physical geodesy'. *Bulletin géodésique* **86**(1), 491–492.

Hellweger, F. and D. Maidment: 1997, 'AGREE-DEM surface reconditioning system'. (October). Online at http://www.ce.utexas.edu/prof/maidment/gishydro/ferdi/research/agree/agree.html (last accessed 2009).

Hengl, T., B. Bajat, D. Blagojevic, and H. I. Reuter: 2008, 'Geostatistical modeling of topography using auxiliary maps'. *Computers & Geosciences* **34**(12), 1886–1899.

Hengl, T., I. S. Evans, H. Tomislav, and I. R. Hannes: 2009, 'Chapter 2 Mathematical and Digital Models of the Land Surface'. In: *Developments in Soil Science*, Vol. 33. Elsevier, pp. 31–63.

Hernández Encinas, L., S. Hoya White, A. Martín del Rey, and G. Rodríguez Sánchez: 2007, 'Modelling forest fire spread using hexagonal cellular automata'. *Applied Mathematical Modelling* **31**(6), 1213–1227.

Heuvelink, G. B. M.: 1998, *Error propagation in environmental modelling with GIS*, Research monographs in geographic information systems. London; Bristol, PA: Taylor & Francis.

Hoffmann, K. and S. Chiang: 1993, *Computational Fluid Mechanics for Engineers, Vol. 1.* Wichita, Kansas: Engineering Educational System.

Höhle, J. and M. Höhle: 2009, 'Accuracy assessment of digital elevation models by means of robust statistical methods'. *ISPRS Journal of Photogrammetry and Remote Sensing* **64**(4), 398–406.

Holland, E. P., J. N. Aegerter, C. Dytham, and G. C. Smith: 2007, 'Landscape as a Model: The Importance of Geometry'. *PLoS Comput Biol* **3**(10), e200.

Holmes, K. W., O. A. Chadwick, and P. C. Kyriakidis: 2000, 'Error in a USGS 30-meter digital elevation model and its impact on terrain modeling'. *Journal of Hydrology* **233**(1-4), 154–173.

Horton, R. E.: 1945, 'Erosional development of streams and their drainage basins; hydrophysical approach to quantitative morphology'. *Geological Society of America Bulletin* **56**(3), 275–370.

Huang, Y. M. and C.-J. Chen: 2009, '3D Fractal reconstruction of terrain profile data based on digital elevation model'. *Chaos, Solitons & Fractals* **40**(4), 1741–1749.

Hutchinson, M., J. Stein, and T. l. . Xu: 2009, 'Locally adaptive gridding of noisy high resolution topographic data'. In: *18th IMACS World Congress MODSIM09 Conference CD*. Cairns, Queensland, Australia.

Hutchinson, M. F.: 1989, 'A New Procedure for Gridding Elevation and Stream Line Data with Automatic Removal of Spurious Pits'. *Journal of Hydrology* **106**(3-4), 211–232.

Hutchinson, M. F.: 2000, 'Optimising the degree of data smoothing for locally adaptive finite element bivariate smoothing splines'. *Australian and New Zealand Industrial and Applied Mathematics Journal* **42**(E), C774–C796.

Hutchinson, M. F. and A. I. Dowling: 1991, 'A continental hydrological assessment of a new grid-based digital elevation model of Australia'. *Hydrol. Process.* **5**(1), 45–58.

Imhof, E.: 2007, *Cartographic Relief Presentation*. Redlands California: ESRI Press.

Intergovernmental Panel On Climate Change: 2014, 'Climate Change 2014 Synthesis Report'.

Isenburg, M., Y. Liu, J. Shewchuk, J. Snoeyink, and T. Thirion: 2006, 'Generating Raster DEM from Mass Points Via TIN Streaming'. In: M. Raubal, H. Miller, A. Frank, and M. Goodchild (eds.): *Lecture Notes in Computer Science*, Vol. 4197. Springer Berlin Heidelberg, pp. 186–198.

Jenny, B. and L. Hurni: 2010, 'Terrain Generalization with multi-scale pyramids constrained by curvature'. In: *A special Joint Symposium of ISPRS Technical Commision IV and AutoCarto*. Orlando Florida.

Jenson, S. K. and J. O. Domingue: 1988, 'Extracting topographic structure from digital elevation data for geographic information-system analysis'. *Photogrammetric Engineering and Remote Sensing* **54**(11), 1593–1600.

Jerri, A. J.: 1977, 'The Shannon sampling theorem - Its various extensions and applications: A tutorial review'. *Proceedings of the IEEE* **65**(11), 1565–1596.

Jin, K.-R. and Y. Wu: 1997, 'Boundary-fitted grid in landscape modeling'. **12**(1), 19–26.

Jones, K. H.: 1998, 'A comparison of algorithms used to compute hill slope as a property of the DEM'. *Computers & Geosciences* **24**(4), 315–323.

Jones, N. L., S. G. Wright, and D. R. Maidment: 1990, 'Watershed Delineation with Triangle-Based Terrain Models'. *Journal of Hydraulic Engineering-ASCE* **116**(10), 1232–1251.

Jones, R.: 2002, 'Algorithms for using a DEM for mapping catchment areas of stream sediment samples'. *Computers & Geosciences* **28**(9), 1051–1060.

Joynes, S.: 2009, 'Get the modelling right first time: A new paradigm in modelling processes'. In: *Water New Zealand Annual Conference*. Rotorua.

Kampf, S. and S. Burges: 2007, 'A framework for classifying and comparing distributed hillslope and catchment hydrologic models'. *Water resources research* **43**(5), W05423.

Kavetski, D. and M. P. Clark: 2010, 'Ancient numerical daemons of conceptual hydrological modeling: 2. Impact of time stepping schemes on model analysis and prediction'. *Water Resour. Res.* **46**(10), W10511.

Kavvas, M.: 1999, 'On the coarse-graining of hydrologic processes with increasing scales'. *Journal of Hydrology* **217**(34), 191 – 202.

Kenny, F. and B. Matthews: 2005, 'A methodology for aligning raster flow direction data with photogrammetrically mapped hydrology'. *Computers & Geosciences* **31**(6), 768–779.

Kenny, F., B. Matthews, and K. Todd: 2008, 'Routing overland flow through sinks and flats in interpolated raster terrain surfaces'. *Computers & Geosciences* **34**(11), 1417–1430.

Klinkenberg, B. and M. F. Goodchild: 1992, 'The Fractal Properties of Topography - a Comparison of Methods'. *Earth Surface Processes and Landforms* **17**(3), 217–234.

Kobbelt, L.: 2000, '$\sqrt{3}$ subdivisionn'. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., pp. 103–112.

Koch, A. and C. Heipke: 2006, 'Integrating 2D Topographic Vector Data with a Digital Terrain Model a Consistent and Semantically Correct Approach'. *Advances in Spatial Data Handling* **7**, 353–364.

Kuczera, G.: 1997, 'Efficient subspace probabilistic parameter optimization for catchment models'. *Water Resources Research* **33**(1), 177–185.

Lacroix, M. P., L. W. Martz, G. W. Kite, and J. Garbrecht: 2002, 'Using digital terrain analysis modeling techniques for the parameterization of a hydrologic model'. *Environmental Modelling & Software* **17**(2), 125–134.

Lamond, J. and E. Penning-Rowsell: 2014, 'The robustness of flood insurance regimes given changing risk resulting from climate change'. *Climate Risk Management* **2**, 1–10.

Landsberg, H. E.: 1981, *The Urban Climate*. Academic Press.

Lea, N. L.: 1992, 'An aspect driven kinematic routing algorithm'. In: A. J. Parsons and A. D. Abrahams (eds.): *Overland Flow: Hydraulics and Erosion Mechanics*. Chapman & Hall, New York.

Ledoux, H. and C. M. Gold: 2008, 'Modelling three-dimensional geoscientific fields with the Voronoi diagram and its dual'. *International Journal of Geographical Information Science* **22**(5), 547–574.

Lee, J.: 1991, 'Comparison of existing methods for building triangular irregular network, models of terrain from grid digital elevation models'. *International journal of geographical information science* **5**(3), 267.

Leopold, L. B. and J. P. Miller: 1956, 'Ephemeral streams-hydraulic factors and their relation to the drainage net'. Technical report, U.S. Geol. Survey. US Government Printing Office.

Li, J. and D. W. Wong: 2009, 'Effects of DEM sources on hydrologic applications'. *Computers, Environment and Urban Systems* **34**(3), 251–261.

Lin, W. T., W. C. Chou, C. Y. Lin, P. H. Huang, and J. S. Tsai: 2008, 'WinBasin: Using improved algorithms and the GIS technique for automated watershed modelling analysis from digital elevation models'. *International Journal of Geographical Information Science* **22**(1), 47–69.

Lindsay, J. B.: 2005, 'The Terrain Analysis System: a toot for hydro-geomorphic applications'. *Hydrological Processes* **19**(5), 1123–1130.

Lindsay, J. B. and I. F. Creed: 2005a, 'Removal of artifact depressions from digital elevation models: towards a minimum impact approach'. *Hydrological Processes* **19**(16), 3113–3126.

Lindsay, J. B. and I. F. Creed: 2005b, 'Sensitivity of digital landscapes to artifact depressions in remotely-sensed DEMs'. *Photogrammetric Engineering and Remote Sensing* **71**(9), 1029–1036.

Lindsay, J. B. and I. F. Creed: 2006, 'Distinguishing actual and artefact depressions in digital elevation data'. *Computers & Geosciences* **32**(8), 1192–1204.

Lindsay, J. B. and M. G. Evans: 2008, 'The influence of elevation error on the morphometrics of channel networks extracted from DEMs and the implications for hydrological modelling'. *Hydrological Processes* **22**(11), 1588–1603.

Lindsay, J. B., J. J. Rothwell, and H. Davies: 2008, 'Mapping outlet points used for watershed delineation onto DEM-derived stream networks'. *Water Resources Research* **44**(8).

Liu, Y., H. Gupta, E. Springer, and T. Wagener: 2008, 'Linking science with environmental decision making: Experiences from an integrated modeling approach to supporting sustainable water resources management'. *Environmental Modelling & Software* **23**(7), 846–858.

Liu, Y. and J. Snoeyink: 2005, 'Flooding Triangulated Terrain'. In: P. Fisher (ed.): *Developments in Spatial Data Handling*. Springer Berlin Heidelberg, pp. 137–148.

Lundmark, A., N. Wadstromer, and H. Li: 1999, 'Recursive subdivisions of the plane yielding nearly hexagonal regions'. *RadioVetenskap och Komunikation*.

MacMillan, R. A., R. K. Jones, and D. H. McNabb: 2004, 'Defining a hierarchy of spatial entities for environmental analysis and modeling using digital elevation models (DEMs)'. *Computers, Environment and Urban Systems* **28**(3), 175–200.

MacMillan, R. A., T. C. Martin, T. J. Earle, and D. H. McNabb: 2003, 'Automated analysis and classification of landforms using high-resolution digital elevation data: Applications and issues'. *Canadian Journal of Remote Sensing* **29**(5), 592–606.

Madsen, H.: 2003, 'Parameter estimation in distributed hydrological catchment modelling using automatic calibration with multiple objectives'. *Advances in Water Resources* **26**(2), 205 – 216.

Magalhàes, S., M. Andrade, W. Randolph Franklin, and G. Pena: 2012, 'A New Method for Computing the Drainage Network Based on Raising the Level of an Ocean Surrounding the Terrain'. In: J. Gensel, D. Josselin, and D. Vandenbroucke (eds.): *Bridging the Geographic Information Sciences*, Lecture Notes in Geoinformation and Cartography. Springer Berlin Heidelberg, pp. 391–407.

Maidment, D.: 2002, *ArcHydro: GIS for Water Resources*. Redlands, CA: ESRI.

Mandelbrot, B. B.: 1967, 'How long is the coast of Britain? Statistical self-similarity and fractional dimension'. *Science* **156**, 636–638.

Mandelbrot, B. B.: 1977, *Fractals: Form, chance, and Dimension*. Freeman.

Mark, D.: 1988, 'Network Models in Geomorphology'. In: *Modelling Geomorphological Systems*. John Wiley and Sons New York., pp. 73–97.

Martz, L. W. and J. Garbrecht: 1999, 'An outlet breaching algorithm for the treatment of closed depressions in a raster DEM'. *Computers and Geosciences* **25**(7), 835–844.

McAllister, M.: 1999, 'A Watershed Algorithm for Triangulated Terrains'. In: *Proc. 11th Canadian Conference on Computational Geometry*. pp. 103–106.

McCormack, J. E. and J. Hogg: 1997, 'Virtual-memory tiling for spatial data handling in GIS'. *Computers & Geosciences* **23**(6), 659–669.

Meadows, D., J. Randers, and D. Meadows: 2004, *Limits to growth: the 30-year update*. White River Junction, Vermont: Chelsea Green Publishing Company.

Meadows, D. H., D. L. Meadows, J. Randers, and W. Behrens III: 1972, *The Limits to Growth*. London: Pan Books.

Meierdiercks, K. L., J. A. Smith, M. L. Baeck, and A. J. Miller: 2010, 'Analyses of Urban Drainage Network Structure and its Impact on Hydrologic Response1'. *Journal of the American Water Resources Association* **46**(5), 932–943.

Meisels, A., S. Raizman, and A. Karnieli: 1995, 'Skeletonizing a DEM into a drainage network'. *Computers & Geosciences* **21**(1), 187–196.

Menduni, G. and V. Riboni: 2000, 'A physically based catchment partitioning method for hydrological analysis'. *Hydrol. Process.* **14**(11-12), 1943–1962.

Meyer, T., M. Eriksson, and R. Maggio: 2001, 'Gradient Estimation from Irregularly Spaced Data Sets'. *Mathematical Geology* **33**(6), 693–717.

Microsoft Corporation: 2009, 'Photosynth'. http://www.photosynth.net/ Accessed 10 March 2011.

Microsoft Corporation: 2013, 'Bing Maps Tile System'. http://msdn.microsoft.com/en-us/library/bb259689.aspx Accessed 20 June, 2013.

Middleton, L. and J. Sivaswamy: 2001, 'Edge detection in a hexagonal-image processing framework'. *Image and Vision Computing* **19**(14), 1071–1081.

Middleton, L. and J. Sivaswamy: 2005a, 'Front matter'. In: *Advances in Pattern Recognition*. Springer London, pp. 71–103.

Middleton, L. and J. Sivaswamy: 2005b, 'The Proposed HIP Framework'. In: *Advances in Pattern Recognition*. Springer London, pp. 27–69.

Miller, C. and R. Laflamme: 1958, 'The digital terrain model - theory and application'. *Photogrammetric Engineering* **24**, 433.

Mitas, L. and H. Mitasova: 1999, 'Spatial interpolation'. *Geographical information systems: principles, techniques, management and applications* **1**, 481–492.

Mizukoshi, H. and M. Aniya: 2002, 'Use of contour-based DEMs for deriving and mapping topographic attributes'. *Photogrammetric Engineering and Remote Sensing* **68,**(1), 83–93.

Moellering, H. and W. Tobler: 1972, 'Geographical Variances'. *Geographical Analysis* **4**(1), 34–50.

Moga, A. N. and M. Gabbouj: 1998, 'Parallel marker-based image segmentation with watershed transformation'. *Journal of Parallel and Distributed Computing* **51**(1), 27–45.

Moore, I. D., E. M. Oloughlin, and G. J. Burch: 1988, 'A Contour Based Topographic Model for Hydrological and Ecological Applications'. *Earth Surface Processes and Landforms* **13**(4), 305–320.

Mori, Y.: 1987, 'Methods for Estimating the Mean and the Standard Deviation of Wind Direction'. *J. Climate Appl. Meteor.* **26**(9), 1282–1284.

Morris, E. M. and D. A. Woolhiser: 1980, 'Unsteady one-dimensional flow over a plane: Partial equilibrium and recession hydrographs'. *Water Resources Research* **16**(2), 355–360.

Muehlenhaus, I.: 2014, *Web Cartography: Map Design for Interactive and Mobile Devices*. Boca Raton: CRC Press.

Mulvaney, T. J.: 1851, 'On the use of self-registering rain and flood gauges in making observations of the relations of rainfall and flood discharges in a given catchment'. *Proceedings of the institution of Civil Engineers of Ireland* **4**(2), 18–33.

Niemann, J. D., R. L. Bras, and D. Veneziano: 2003, 'A physically based interpolation method for fluvially eroded topography'. *Water Resour. Res.* **39**(1), 1017.

Nobel Media AB: 2014, 'The 2014 Nobel Prize in Physiology or Medicine - Press Release'. http://www.nobelprize.org/nobel_prizes/medicine/laureates/2014/press.html Accessed 11 Dec 2016.

O'Callaghan, J. F. and D. M. Mark: 1984, 'The extraction of drainage networks from digital elevation data'. *Computer Vision, Graphics, and Image Processing* **28**(3), 323–344.

Ohya, T., M. Iri, and K. Murota: 1984, 'A fast Voronoi-diagram algorithm with quaternary tree bucketing'. *Information Processing Letters* **18**(4), 227–231.

Okabe, A., B. N. Boots, and K. Sugihara: 1992, *Spatial tessellations: concepts and applications of Voronoi diagrams*, Wiley series in probability and mathematical statistics. Chichester, England ;New York: Wiley & Sons.

Oksanen, J. and T. Sarjakoski: 2005, 'Error propagation of DEM-based surface derivatives'. *Computers & Geosciences* **31**(8), 1015–1027.

Olivera, F., J. Famiglietti, and K. Asante: 2000, 'Global-scale flow routing using a source-to-sink algorithm'. *Water Resources Research* **36**(8), 2197–2207.

Oloughlin, E. M.: 1986, 'Prediction of Surface Saturation Zones in Natural Catchments by Topographic Analysis'. *Water Resources Research* **22**(5), 794–804.

Oreskes, N. and K. Belitz: 2001, 'Philosophical issues in model assessment'. *Model validation: Perspectives in hydrological science* **23**.

Orlandini, S. and G. Moretti: 2009, 'Determination of surface flow paths from gridded elevation data'. *Water Resour. Res.* **45**(3), W03417.

Orlandini, S., G. Moretti, M. Franchini, B. Aldighieri, and B. Testa: 2003, 'Path-based methods for the determination of nondispersive drainage directions in grid-based digital elevation models'. *Water Resources Research* **39**(6).

Overgaard, J., D. Rosbjerg, and M. B. Butts: 2006, 'Land-surface modelling in hydrological perspective? a review'. *Biogeosciences* **3**(2), 229–241.

Pajarola, R.: 2002, 'Overview of Quadtree-based Terrain Triangulation and Visualization'. UCI-ICS Technical Report No. 02-01, Computer Graphics Lab Information & Computer Science Department University of California Irvine.

Palacios-Vélez, O. and B. Cuevas-Renaud: 1986, 'Automated river-course, ridge and basin delineation from digital elevation data'. *Journal of Hydrology* **86**(3-4), 299–314.

Pan, F. F., C. D. Peters-Lidard, M. J. Sale, and A. W. King: 2004, 'A comparison of geographical information systems-based algorithms for computing the TOPMODEL topographic index'. *Water Resources Research* **40**(6).

Panday, S. and P. S. Huyakorn: 2004, 'A fully coupled physically-based spatially-distributed model for evaluating surface/subsurface flow'. *Advances in Water Resources* **27**(4), 361 – 382. A Tribute to George F. Pinder.

Peckham, S. D., H. Tomislav, and I. R. Hannes: 2009, 'Chapter 25 Geomorphometry and Spatial Hydrologic Modelling'. In: *Developments in Soil Science*, Vol. Volume 33. Elsevier, pp. 579–602.

Peucker, T. K. and D. H. Douglas: 1975, 'Detection of Surface-Specific Points by Local Parallel Processing of Discrete Terrain Elevation Data'. *Computer Graphics and Image Processing* **4**(4), 375–387.

Pike, R. J.: 2002, 'A Bibliography of Terrain Modeling (Geomorphometry), the Quantitative Representation of Topography - Supplement 4.0. Open-File Report 02-465'. Technical report, United States Geological Survey, Menlo Park, California.

Pike, R. J. and K.-L. Kimberly: 2005, 'Digital Terrain Modeling'. In: *Encyclopedia of Social Measurement*. New York: Elsevier, pp. 669–675.

Pirotti, F. and P. Tarolli: 2010, 'Suitability of LiDAR point density and derived landform curvature maps for channel network extraction'. *Hydrological Processes* **24**, 1187–1197.

Planchon, O. and F. Darboux: 2002, 'A fast, simple and versatile algorithm to fill the depressions of digital elevation models'. *CATENA* **46**(2-3), 159–176.

Platings, M. and A. M. Day: 2004, 'Compression of Large-Scale Terrain Data for Real-Time Visualization Using a Tiled Quad Tree'. *Computer Graphics Forum* **23**(4), 741–759.

Ponce, V. M.: 1978, 'Applicability of Kinematic and Diffusion Models'. *Journal of the Hydraulics Division* **104**(3), 353–360.

Priyadarshi, B.: 2007, 'Voronoi diagram in optimal path planning'. In: L. G. Marina (ed.): *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)*. pp. 38–47.

Puppo, E.: 1998, 'Variable resolution triangulations'. *Computational Geometry* **11**(3-4), 219–238.

Purss, M. B. J., R. Gibb, F. Samavati, P. Peterson, and J. Ben: 2016, 'The OGC ®; Discrete Global Grid System core standard: A framework for rapid geospatial integration'. In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. pp. 3610–3613.

Quinn, P., K. Beven, P. Chevallier, and O. Planchon: 1991, 'The Prediction of Hillslope Flow Paths for Distributed Hydrological Modeling Using Digital Terrain Models'. *Hydrological Processes* **5**(1), 59–79.

Quinn, P. F., K. J. Beven, and R. Lamb: 1995, 'The in(a/tan2) index: How to calculate it and how to use it within the topmodel framework'. *Hydrological Processes* **9**, 161–182.

Raaflaub, L. D. and M. J. Collins: 2006, 'The effect of error in gridded digital elevation models on the estimation of topographic parameters'. *Environmental Modelling & Software* **21**(5), 710–732.

Refshaard, J. C., B. Storm, and V. P. Singh: 1995, 'MIKE SHE'. *Computer models of watershed hydrology* pp. 809–846.

Reggiani, P. and J. Schellekens: 2003, 'Modelling of hydrological responses: the representative elementary watershed approach as an alternative blueprint for watershed modelling'. *Hydrological Processes* **17**(18), 3785–3789.

Reinoso, J.: 2010, 'A priori horizontal displacement (HD) estimation of hydrological features when versioned DEMs are used'. *Journal of Hydrology* **384**(1-2), 130–141.

Renssen, H. and J. M. Knoop: 2000, 'A global river routing network for use in hydrological modeling'. *Journal of Hydrology* **230**(3-4), 230–243. 5.

Ritter, P.: 1987, 'A vector-based slope and aspect generation algorithm'. *Photogrammetric Engineering and Remote Sensing* **53**(8), 1109–1111.

Robinson, W. S.: 1950, 'Ecological Correlations and the Behavior of Individuals'. *American Sociological Reviews* **15**, 351–357.

Roerdink, J. and A. Meijster: 2000, 'The Watershed Transform: Definitions, Algorithms and Parallelization Stratergies'. *Fundamenta Informica* **40**, 187–228.

Romano-Díaz, E. and R. Van De Weygaert: 2007, 'Delaunay Tessellation Field Estimator analysis of the PSCz local Universe: density field and cosmic flow'. *Monthly Notices of the Royal Astronomical Society* **382**(1), 2–28.

Ros, D. D. and M. Borga: 1997, 'Use of digital elevation model data for the derivation of the geomorphological instantaneous unit hydrograph'. *Hydrological Processes* **11**(1), 13–33.

Rossman, L. A.: 2009, 'Storm Water Management Model (SWMM) User's Manual'. U.S. Environmental Protection Agency, Cincinnati, OH 45268, version 5.0 edition.

Sahr, K., D. White, and A. J. Kimerling: 2003, 'Geodesic Discrete Global Grid Systems'. *Cartography and Geographic Information Science* **30**(2), 121–134.

Saltelli, A., K. Chan, and E. M. Scott (eds.): 2000, *Sensitivity Analysis*, series in probability and statistics. Chichester: John Wiley & Sons.

Samet, H.: 1984, 'The Quadtree and Related Hierarchical Data Structures'. *ACM Comput. Surv.* **16**(2), 187–260.

Sasowsky, I. D.: 2006, 'Model verification and documentation are needed'. *Eos, Transactions American Geophysical Union* **87**(25), 248–248.

Satellite Imaging Corporation: 2014, 'WorldView3 data sheet'. https://www.spaceimagingme.com/downloads/sensors/datasheets/DG_WorldView3_DS_2014.pdf Downloaded May 2017.

Saunders, W. K.: 2000, 'Preparation of DEMs for Use in Environmental Modeling Analysis'. In: D. Maidment and D. Djokic (eds.): *Hydrologic and Hydraulic Modeling Support with Geographic Information Systems*. Redlands: CA: ESRI Press, pp. 22–51.

Scarlatos, L. and T. Pavlidis: 1992, 'Hierarchical triangulation using cartographic coherence'. *CVGIP: Graphical Models and Image Processing* **54**(2), 147–161.

Schäppi, B., P. Perona, P. Schneider, and P. Burlando: 2010, 'Integrating river cross section measurements with digital terrain models for improved flow modelling applications'. *Computers & Geosciences* **36**(6), 707–716.

Schneider, B.: 2002, 'Uncertainty of Local Form in Digital Terrain Modelling'. In: *Proc. GIS Research UK 9th Annual Conference*.

Schumann, G., P. D. Bates, M. S. Horritt, P. Matgen, and F. Pappenberger: 2009, 'Progress in integration of remote sensing derived flood extent and stage data and hydraulic models'. *Rev. Geophys.* **47**(4).

Seibert, J. and B. L. McGlynn: 2007, 'A new triangular multiple flow direction algorithm for computing upslope areas from gridded digital elevation models'. *Water Resour. Res.* **43**(4), W04501.

Shamos, M. I.: 1975, 'Closest-point problems'. In: D. Hoey (ed.): *Annual IEEE Symposium on Foundations of Computer Science*. pp. 151–162.

Shary, P. A., L. S. Sharaya, and A. V. Mitusov: 2002, 'Fundamental quantitative methods of land surface analysis'. *Geoderma* **107**(1-2), 1–32.

Shreve, R. L.: 1966, 'Statistical law of stream numbers'. *The Journal of Geology* **74**(1), 17–37.

Sicilia, A. and F. Judice: 2010, 'Lattice methods for fluid animation in games'. *J Comput. Entertain.* **7**(4), 1–29.

Silveira, R. and R. van Oostrum: 2007, 'Flooding Countries and Destroying Dams'. In: F. Dehne, J.-R. Sack, and N. Zeh (eds.): *Lecture Notes in Computer Science*, Vol. 4619. Springer Berlin / Heidelberg, pp. 227–238.

Singh, V. P. and D. A. Woolhiser: 2002, 'Mathematical Modeling of Watershed Hydrology.'. *Journal of Hydrologic Engineering* **7**(4), 270–292.

Skidmore, A.: 2002, 'Taxonomy of environmental models in the spatial sciences'. In: A. Skidmore (ed.): *Environmental modelling with GIS and remote sensing*. United Kingdom: Taylor & Francis, pp. 8–24.

Smith, J., B. Phillips, and S. Yu: 2006, 'Modelling Overland Flows and Drainage Augmentations in Dubbo'. In: *46th Floodplain Management Authorities Conference*. Lismore.

Smith, S. L., D. A. Holland, and P. A. Longley: 2005, 'Quantifying Interpolation Errors in Urban Airborne Laser Scanning Models'. *Geographical Analysis* **37**(2), 200–224.

Snavely, N., S. M. Seitz, and R. Szeliski: 2006, 'Photo Tourism: Exploring image collections in 3D. ACM Transactions on Graphics'. In: *Proceedings of SIGGRAPH 2006*.

Snyder, W. E., H. Qi, and W. Sander: 1999, 'A coordinate system for hexagonal pixels'. In: *Proceedings of the 1999 Medical Imaging - Image Processing*. San Diego, CA, USA.

Soille, P.: 2004, 'Optimal removal of spurious pits in grid digital elevation models'. *Water Resources Research* **40**(12), W12509: 1–9.

Soille, P., J. Vogt, and R. Colombo: 2003, 'Carving and adaptive drainage enforcement of grid digital elevation models'. *Water Resources Research* **39**(12), 10: 1–13.

Spear, R. C., T. M. Grieb, and N. Shang: 1994, 'Parameter uncertainty and interaction in complex environmental models'. *Water Resources Research* **30**(11), 3159–3169.

Stelling, G. and M. Zijlema: 2003, 'An accurate and efficient finite-difference algorithm for non-hydrostatic free-surface flow with application to wave propagation'. *International Journal for Numerical Methods in Fluids* **43**(1), 1–23.

Strahler, A. N.: 1952, 'Dynamic basis of geomorphology'. *Geological Society of America Bulletin* **63**(9), 923–938.

Tarboton, D. G.: 1997, 'A new method for the determination of flow directions and upslope areas in grid digital elevation models'. *Water Resources Research* **33**(2), 309–319.

Tarboton, D. G.: 2008, 'Terrain Analysis Using Digital Elevation Models (TauDEM)'. http://hydrology.neng.usu.edu/taudem/ Accessed March 2011.

Tarboton, D. G., R. L. Bras, and I. Rodriguezituribe: 1991, 'On the Extraction of Channel Networks from Digital Elevation Data'. *Hydrological Processes* **5**(1), 81–100.

Tate, E. C., D. R. Maidment, F. Olivera, and D. J. Anderson: 2002, 'Creating a Terrain Model for Floodplain Mapping:'. *Journal of Hydrologic Engineering* **7**(2), 100.

Temme, A. J. A. M., J. M. Schoorl, and A. Veldkamp: 2006, 'Algorithm for dealing with depressions in dynamic landscape evolution models'. *Computers & Geosciences* **32**(4), 452–461.

Terrafirma: 2011, 'International Geohazard Information Service'. Accessed April 2017.

Thompson, J. F., Z. U. Warsi, and C. W. Mastin: 1985, *Numerical grid generation: foundations and applications*, Vol. 45. North-Holland Amsterdam.

Tobler, W. R.: 1970, 'A computer movie simulating urban growth in the Detroit Region'. *Economic Geography* **46**, 234–240.

Todini, E.: 1988, 'Rainfall-runoff modeling - Past, present and future'. *Journal of Hydrology* **100**(1), 341 – 352.

Tse, R. O., C. Gold, and D. Kidner: 2007, 'Using the Delaunay Triangulation/ Voronoi Diagram to extract Building Information from Raw LIDAR Data'. In: G. Chris and K. Dave (eds.): *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)*. pp. 222–229.

Tsubaki, R. and I. Fujita: 2010, 'Unstructured grid generation using LiDAR data for urban flood inundation modelling'. *Hydrol. Process.* **24**(11), 1404–1420.

Tsui, P. H. Y. and A. J. Brimicombe: 1997, 'Adaptive Recursive Tessellations (ART) for Geographical Information Systems'. *International Journal of Geographical Information Science* **11**(3), 247–263.

Tucker, G. E., S. T. Lancaster, N. M. Gasparini, R. L. Bras, and S. M. Rybarczyk: 2001, 'An object-oriented framework for distributed hydrologic and geomorphic modeling using triangulated irregular networks'. *Computers & Geosciences* **27**(8), 959–973.

Turcotte, R., J. P. Fortin, A. N. Rousseau, S. Massicotte, and J. P. Villeneuve: 2001, 'Determination of the drainage structure of a watershed using a digital elevation model and a digital river and lake network'. *Journal of Hydrology* **240**(3-4), 225–242.

United Nations: 2010, *World Urbanisation prospects: the 2009 revision*. New York.

Valle, Y. and J. Ortiz: 2011, 'An efficient representation of quadtrees and bintrees for multiresolution terrain models'. *Geo-spatial Information Science* **14**(3), 198–206.

Van De Ville, D., T. Blu, M. Unser, W. Philips, I. Lemahieu, and R. Van de Walle: 2004, 'Hex-splines: a novel spline family for hexagonal lattices'. *Image Processing, IEEE Transactions on* **13**, 758–772.

Van De Ville, D., W. Philips, and I. Lemahieu: 2002, 'Least-squares spline resampling to a hexagonal lattice'. *Signal Processing: Image Communication* **17**(5), 393–408.

Van Niel, K. and S. W. Laffan: 2003, 'Gambling with randomness: the use of pseudo-random number generators in GIS'. *International Journal of Geographical Information Science* **17**(1), 49–68.

Vázquez, R. and J. Feyen: 2007, 'Assessment of the effects of DEM gridding on the predictions of basin runoff using MIKE SHE and a modelling resolution of 600 m'. *Journal of Hydrology* **334**(1-2), 73–87.

Vincent, L. and P. Soille: 1991, 'Watersheds in Digital Spaces - an Efficient Algorithm Based on Immersion Simulations'. *Ieee Transactions on Pattern Analysis and Machine Intelligence* **13**(6), 583–598.

Vivoni, E. R., V. Y. Ivanov, R. L. Bras, and D. Entekhabi: 2004, 'Generation of triangulated irregular networks based on hydrological similarity'. *Journal of Hydrologic Engineering* **9**(4), 288–302.

Vivoni, E. R., V. Y. Ivanov, R. L. Bras, and D. Entekhabi: 2005a, 'On the effects of triangulated terrain resolution on distributed hydrologic model response'. *Hydrological Processes* **19**(11), 2101–2122.

Vivoni, E. R., V. Teles, V. Y. Ivanov, R. L. Bras, and D. Entekhabi: 2005b, 'Embedding landscape processes into triangulated terrain models'. *International Journal of Geographical Information Science* **19**(4), 429–457.

Vogt, J. V., R. Colombo, and F. Bertolo: 2003, 'Deriving drainage networks and catchment boundaries: a new methodology combining digital elevation data and environmental characteristics'. *Geomorphology* **53**(3-4), 281–298.

Von Herzen, B. and A. H. Barr: 1987, 'Accurate Triangulations of Deformed, Intersecting Surfaces'. *SIGGRAPH Comput. Graph.* **21**(4), 103–110.

Walker, J. P. and G. R. Willgoose: 1999, 'On the Effect of Digital Elevation Model Accuracy on Hydrology and Geomorphology'. *Water Resour. Res.* **35**(7), 2259–2268.

Wang, J.-P. and Q. Liang: 2011, 'Testing a new adaptive grid-based shallow flow model for different types of flood simulations'. *Journal of Flood Risk Management* **4**(2), 96–103.

Wang, K., C.-P. Lo, G. A. Brook, and H. R. Arabnia: 2001, 'Comparison of existing triangulation methods for regularly and irregularly spaced height fields'. *International Journal of Geographical Information Science* **15**(8), 743–762.

Wang, L. and H. Liu: 2006, 'An efficient method for identifying and filling surface depressions in digital elevation models for hydrologic analysis and modelling'. *International Journal of Geographical Information Science* **20**(2), 193–213.

Wang, X. and Z.-Y. Yin: 1998, 'A comparison of drainage networks derived from digital elevation models at two scales'. *Journal of Hydrology* **210**(1-4), 221–241.

Watson, D. F.: 1992, *Contouring: A Guide to the Analysis and Display of Spatial Data*. Pergamon.

Weidner, U. and W. Förstner: 1995, 'Towards automatic building extraction from high-resolution digital elevation models'. *ISPRS Journal of Photogrammetry and Remote Sensing* **50**(4), 38–49.

Wicks, J. and J. Bathurst: 1996, 'SHESED: a physically based, distributed erosion and sediment yield component for the {SHE} hydrological modelling system'. *Journal of Hydrology* **175**(1), 213 – 238.

Williams, J. R. and W. V. LaSeur: 1976, 'Water yield model using SCS curve numbers'. *Journal of the Hydraulics Division* **102**(9), 1241–1253.

Wilson, J. P., G. Aggett, D. Yongxin, and C. S. Lam: 2008, 'Water in the Landscape: A Review of Contemporary Flow Routing Algorithms'. In: Q. Zhou, B. Lees, and G.-a. Tang (eds.): *Advances in Digital Terrain Analysis*, Vol. Section 3 of *Lecture Notes in Geoinformation and Cartography*. Springer Berlin Heidelberg, pp. 213–236.

Wilson, J. P., H. Mitasova, and D. J. Wright: 2000, 'Water resource applications of Geographic Information Systems'. *Journal of the Urban and Regional Information Systems Association* **12**, 61–79.

Winter, S. and A. Frank: 2000, 'Topology in Raster and Vector Representation'. *GeoInformatica* **4**(1), 35–65.

Wise, S.: 2000, 'Assessing the quality for hydrological applications of digital elevation models derived from contours'. *Hydrological Processes* **14**(11-12), 1909–1929.

Wolock, D. M. and G. J. Mccabe: 1995, 'Comparison of Single and Multiple Flow Direction Algorithms for Computing Topographic Parameters in Topmodel'. *Water Resources Research* **31**(5), 1315–1324.

Wood, J.: 1998, 'Modelling the continuity of surface form using DEMs'. In: T. K. Poiker and N. R. Chrisman (eds.): *In The 8th International Symposium on Spatial Data Handling*. Vancouver, Canada, pp. 725–736.

Wood, J., H. Tomislav, and I. R. Hannes: 2009, 'Chapter 10 Overview of Software Packages Used in Geomorphometry'. In: *Developments in Soil Science*, Vol. Volume 33. Elsevier, pp. 257–267.

Worboys, M.: 1998, 'Imprecision in Finite Resolution Spatial Data'. **2**(3), 257–279.

Wright, J. W.: 2017, 'RHSM-PhD'. Github. Accessed July 2017.

Wright, J. W. and G. H. Leonard: 2012, 'Evaluation of automated flow direction algorithms for defining urban stormwater catchment boundaries'. *New Zealand Surveyor*.

Wright, J. W., A. B. Moore, and G. H. Leonard: 2014, 'Flow direction algorithms in a Hierarchical Hexagonal Surface Model'. *Journal of Spatial Science* pp. 1–14.

Wu, J. and K. Amaratunga: 2003, 'Wavelet triangulated irregular networks.'. *International Journal of Geographical Information Science* **17**(3), 273–289.

Yang, Q., T. R. McVicar, T. G. Van Niel, M. F. Hutchinson, L. Li, and X. Zhang: 2007, 'Improving a digital elevation model by reducing source data errors and optimising interpolation algorithm parameters: An example in the Loess Plateau, China'. *International Journal of Applied Earth Observation and Geoinformation* **9**(3), 235–246.

Yeh, G.-T., H.-P. Cheng, J.-R. Cheng, H.-C. J. Lin, and W. D. Martin: 1998, 'A Numerical Model Simulating Water Flow and Contaminant and Sediment Transport in WAterSHed

Systems of 1-D Stream-River Network, 2-D Overland Regime, and 3-D Subsurface Media (WASH123D: Version 1.0)'. Final report, Army Engineer Waterways Experiment Station Vicksburg MS Coastal Hydraulics Lab.

Yue, T.-X., Z.-P. Du, D.-J. Song, and Y. Gong: 2007, 'A new method of surface modeling and its application to DEM construction'. *Geomorphology* **91**(1-2), 161–172.

Zandbergen, P. A.: 2010, 'Accuracy Considerations in the Analysis of Depressions in Medium-Resolution Lidar DEMs'. *Giscience & Remote Sensing* **47**(2), 187–207.

Zegeling, P.: 2004, 'Tensor-product adaptive grids based on coordinate transformations'. *Journal of Computational and Applied Mathematics* **166**(1), 343 – 360.

Zhang, J., P. Atkinson, and M. Goodchild: 2014, *Scale in Spatial Information and Analysis*. Boca Raton: CRC Press.

Zhu, H., J. R. Eastman, and J. Toledano: 2001, 'Triangulated irregular network optimization from contour data using bridge and tunnel edge removal'. *International Journal of Geographical Information Science* **15**(3), 271–286.