Improved numerical methods for distributed hydrological models

by

Andrew P. Snowdon

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Civil Engineering

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Distributed hydrological models have been used for decades to calculate and predict the movement of water and energy within watersheds. These models have evolved from relatively simple empirical applications into complex spatially distributed and physically-based programs. However, the evolution of distributed hydrological models has not involved the improvement of the numerical methods used to calculate the redistribution of water and energy in the watershed. Because of this, many models still use numerical methods that are potentially inaccurate.

In order to simulate the transport of water and energy in a hydrological model, typical numerical methods employ an operator splitting approach. Operator splitting (OS) essentially breaks down the set of coupled ordinary differential equations (ODEs) that define a hydrological model into separate ODEs that can be solved individually. The dominant operator splitting method in surface water models is the ordered series approach. Because the ordered series approach treats parallel hydrological processes as if they happen in series, it is prone to errors that can significantly reduce the accuracy of model results. The impact that operator splitting errors have upon hydrologic model results is, to date, unknown.

Using a new distributed hydrological model, Raven, the impact of operator splitting errors is investigated. Understanding these errors will lead to better numerical methods for reducing errors in models and to shed light on the shortcomings of hydrological models with respect to numerical method choice. Alternative numerical methods - the explicit Euler and the implicit iterative Heun methods - are implemented and assessed in their ability to minimize errors and produce more accurate distributed hydrological models.

Acknowledgements

First, I would like to thank my supervisor, James R. Craig. You have given me one of the most valuable experiences of my life and I will always appreciate that. Your dedication to your craft and your never failing ability to challenge me to push beyond the expected has opened my eyes and my mind to new possibilities. Thank you.

Thank you to the exceptional professors in the Department of Civil and Environmental Engineering who have always made me feel welcome, encouraged me, challenged me and been excited to share their knowledge with me. In particular I would like to thank Ric Soulis, Bryan Tolson, Jon Sykes, Shawn Matott and Bill Quinton. You have all contributed to my experiences and helped me succeed.

To my office mates, thank you for being the wonderful supportive people that you are. You have all been sounding boards, shared the pain and made me laugh (a necessity for grad work). Thank you Trish, Wayne, Angela, Sanders, Lucy, Frank and Ben.

Most importantly, I want to thank my family. Without you, I wouldn't be who I am or becoming who I want to be. Thank you mom, your never ceasing line of questioning has forced me to better understand my own work and your concern that I don't eat enough has provided me with more free meals that I can imagine. Dad, your respect and support have always kept me on track and your words of wisdom have always help point me in the right direction. Thank you! Rick, my brother, without you, I would have never made it anywhere in life. I am always trying to catch up to you, you motivate me to go beyond what I think is possible. Torie, my little sister, you are the person who keeps me off balance and forces me to rethink my actions. This has always forced me to plan, replan, aim high and then aim higher. Gotta love sibling rivalry!

Lastly, I would to thank my friends who have helped me de-stress, unwind and generally forget about my research (even if it is only for a short time). There are too many of you to list individually, so I will just say thank you to you all!

Dedication

This thesis is dedicated to Leslie, my Ms. Wright. Without you and your never ending patience and support, this would never have been completed.

Thank you!

Contents

List of Figures

List of Tables

List of Symbols

α_b	Bucket baseflow constant [-]
α_V	VIC alpha coefficient [-]
α_{PT}	Priestley-Taylor coefficient [-]
Δt	Timestep, Change in time [T]
δ_t	Difference between mean monthly maximum and mean monthly minimum temperature $[\theta]$
$\frac{\partial S}{\partial t}\Big\|_x$	Water/ energy storage variable rate of change over time due to process 'x' $[LT^{-1}]$
γ	Psychrometric constant (0.066 KPaK^{-1})
γ_V	VIC evaporation coefficient [-]
γ_{V2}	VIC runoff coefficient [-]
λ_f	Latent heat of fusion (334 $KJKg^{-1}$)
λ_n	Mean of the power transformed topographic index [-]
λ_v	Latent heat of vaporization (2260 $KJKg^{-1}$)
ϕ_i	Water content of storage unit $[L^3L^{-3}]$
ψ	Soil porosity $[L^3L^{-3}]$
ρ_a	Density of air $[ML^{-3}]$
ρ_w	Density of water $[ML^{-3}]$
ξ_{MB}	Mass balance error $[L]$
A	Drainage basin area $[L^2]$
A_i	Generic model storage unit $[L]$

A_{ch} Channel area $[L^2]$

A_{sat} Saturated area [-]

b VIC exponent for surface runoff [-]

B_j Generic model storage unit $[L]$

c VIC FUSE percolation exponent [-]

C_1 Muskingum coefficient 1 [-]

C_2 Muskingum coefficient 2 [-]

C_3 Muskingum coefficient 3 [-]

c_a Heat capacity of air ($1.00\text{x}10^{-3} MJKg^{-1}K^{-1}$)

C_D Canopy drip proportion [-]

c_r Rational method conversion factor

C_t Temperature reduction coefficient (a function of relative humidity) [-]

C_{at} Atmospheric Conductance $[LT^{-1}]$

C_{can} Canopy Conductance $[LT^{-1}]$

C_{cap} Canopy capacity $[L]$

C_{stor} Canopy storage for covered areas $[L]$

e Vapour pressure $[P]$

e_1 Evaporation rate from upper soil layer $[LT^{-1}]$

e_2 Evaporation rate from lower soil layer $[LT^{-1}]$

E_c Constant evaporation rate $[LT^{-1}]$

e_v Efficiency of vertical transport of water vapour $[LT^2M^{-1}]$

e_{ht} Height of vapour pressure measurement $[L]$

E_{ow} Open Water Evaporation $[LT^{-1}]$

e_{sat} Saturated vapour pressure $[P]$

F_c Fraction of canopy coverage [-]

F_i Accumulated infiltration $[L]$

f_n Function of x and y using initial values or solution from previous timestep

F_t Trunk Fraction [-]

f_{n+1}^P Function of x and y using predictor approximation values or solution from previous timestep

F_{PET} Fraction of PET that is open water evaporation [-]

G Soil heat flux $[LT^{-1}]$

I River inflow rate $[L^3T^{-1}]$

I_1 Theoretical hydrological process

I_2 Theoretical hydrological process

I_3 Theoretical hydrological process

I_4 Theoretical hydrological process

I_r Rainfall Intensity $[LT^{-1}]$

K Shortwave Radiation $[EL^{-2}T^{-1}]$

k Von Karman constant [-]

k_1 VIC runoff coefficient [-]

K_e Effective hydraulic conductivity $[LT^{-1}]$

K_m Muskingum proportionality coefficient [-]

k_s Potential baseflow rate $[LT^{-1}]$

k_u Potential percolation rate $[LT^{-1}]$

L Longwave Radiation $[EL^{-2}T^{-1}]$

M_a Melt Factor $[LT^{-1}\theta^{-1}]$

M_C Canopy Drainage $[LT^{-1}]$

M_D Canopy Drip $[LT^{-1}]$

M_n Manning coefficient [-]

M_r Restricted melt factor (2.0 $mm\ day^1\ {}^\circ C^{-1}$)

m_{et} Slope of the relation between the saturated vapour pressure and temperature $[PK^{-1}]$

M_{ij} The combined rates of one or more hydrological processes that move water from the i^{th} storage unit to the j^{th} storage unit $[LT^{-1}]$

mn Maximum soil water content of lower soil layer $[L]$

n VIC FUSE baseflow exponent [-]

NS Number of state variables [-]

P The physical and empirical parameters that define an HRU

P_c Partition coefficient $[LT^{-1}]$

P_p Precipitation $[LT^{-1}]$

P_{ch} Channel perimeter $[L]$

P_{cum} Cumulative precipitation $[L]$

PET Potential evapotranspiration rate $[LT^{-1}]$

Q An external flux to or from an HRU $[L]$

q_b Baseflow rate $[LT^{-1}]$

q_b Percolation rate $[LT^{-1}]$

Q_o River outflow rate $[L^3T^{-1}]$

q_p Percolation rate $[LT^{-1}]$

Q_{cum} Cumulative outflow from system $[L]$

q_{if} Interflow rate $[LT^{-1}]$

q_{sfof} Overflow rate from lower soil layer $[LT^{-1}]$

q_{sx} Surface runoff $[LT^{-1}]$

q_{ufof} Overflow rate from upper soil layer $[LT^{-1}]$

R Rational method runoff coefficient [-]

r Retention Parameter $[L]$

r_1 Relative root fraction of upper soil layer [-]

r_2 Relative root fraction of lower soil layer [-]

R_a Total incoming radiation $[EL^{-2}T^{-1}]$

R_m River slope [-]

R_n Net radiation $[EL^{-2}T^{-1}]$

R_s Snow roughness $[L]$

S_0 State variable value at beginning of timestep $[L]$

S_1 Soil water content of upper soil layer $[L]$

S_1^T Tension water content in upper soil layer $[L]$

S_2 Soil water content in lower soil layer $[L]$

S_2^T Tension water content in lower soil layer $[L]$

S_f Effective suction at wetting front $[L]$

$S_{1,max}$ Maximum soil water content of upper soil layer $[L]$

$S_{1,max}^T$ Maximum tension storage in upper soil layer $[L]$

$S_{2,max}$ Maximum soil water content in lower soil layer $[L]$

$S_{2,max}^T$ Maximum tension storage in lower soil layer $[L]$

$S_{c,max}$ Maximum canopy storage capacity for covered areas $[L]$

S_{crit} VIC critical storage coefficient [-]

S_{init} Initial system mass/ energy storage $[L]$

S_{max} VIC coefficient maximum [-]

S_{sat} VIC coefficient [-]

t Time $[T]$

T_a Mean temperature of air $[\theta]$

T_f Freezing temperature $[\theta]$

t_t Reach travel time $[T]$

$T_{(avg\ d)}$ Mean temperature in timestep $[\theta]$

T_{sw} Sum of all mass/ energy in system $[L]$

v Wind Speed $[LT^{-1}]$

v_{ht} Height of wind velocity measurement $[L]$

vol Volume of water in reach at end of timestep $[L^3T^{-1}]$

w_α Relative Humidity [-]

X Muskingum weighing coefficient [-]

y_n Initial value or solution from previous timestep

y_{n+1} Approximation of exact solution at current timestep

y_{n+1}^{C} Corrector approximation of the exact solution

y_{n+1}^{P} Predictor approximation of the exact solution

Chapter 1

Introduction

1.1 Motivation

Distributed hydrological models are used for water resource management, climate prediction, storm response design and for many other purposes. For water management decisions to be based on the best possible information, these models need to be both accurate and computationally efficient. While the accuracy of model results is often in question, the accuracy of the numerical methods behind the models is rarely discussed. Attention is instead focused upon the ability of physical process algorithms (e.g., equations describing evaporation, infiltration, percolation, etc.) to represent nature. There is a need to improve our understanding of how the choice of numerical method impacts the accuracy of hydrological models and a need to improve the numerical methods used in distributed hydrological models.

Many distributed hydrological models are thought to improve in accuracy when more detailed and/or physically appropriate process algorithms (e.g., for evaporation or snowmelt) are included in the model. These advanced subprocess models are typically more computationally intense than simpler subprocess models. Their implementation has been made possible, in part, due to advancements in computer hardware speed. While these more robust physical process algorithms are thought to improve the capabilities of the model, little effort has been expended to likewise improve the core of the model: the global mathematical solvers that connect the individual process algorithms together. Most models either (1) still use mathematical solvers that were designed to be used for a simpler set of subprocesses or (2) are assembled without regard for what the most appropriate numerical method

may be. In addition, the majority of models have hard-coded their mathematical solvers inside of physical process algorithms, entwining the solvers and processes and making it extremely difficult to change numerical solver. This difficulty is a major hurdle to overcome when attempting to understand and improve numerical models.

A primary goal of this research is to develop an understanding of the shortcomings of the conventional numerical methods used to solve distributed hydrological models. This is a poorly understood topic which has received little or no attention in the research literature. An objective of this research to demonstrate how certain mathematical solvers can be improved to increase the accuracy and efficiency of distributed hydrological models. Through this thesis, it will be shown that the use of alternative numerical solvers may minimize errors and improve the numerical accuracy of distributed hydrological models.

1.2 Research Overview

Hydrological models are composed of sets of coupled ordinary differential equations (ODEs) and partial differential equations (PDEs), used to describe the water balance and energy balance within a watershed. Numerical methods are required to solve these systems of coupled ODEs and PDEs. Most hydrologic models currently apply a method of numerical solution that relies upon operator splitting (OS). The most conventional OS method, the ordered series approach, uses a technique whereby hydrological processes that actually occur in parallel are numerically treated as if they occur in series. The 'in series' treatment leads to numerical errors known as operator splitting errors. Operator splitting errors can impart significant inaccuracies in model results and lead to large discrepancies that can compound over time. These inaccuracies have been discussed in groundwater transport literature [Jacques et al., 2006; Simpson and Landman, 2008; Kanney et al., 2003; Valocchi and Malmstead, 1992; Carrayrou et al., 2004] but have been mostly neglected in surface water literature. There is a need to understand the impact of errors produced by operator splitting methods and how those errors affect the trustworthiness of model results.

Through the design and construction of the modelling software Raven [Craig and Snowdon, 2010], the ability to test numerical solvers used within distributed hydrological models has become much simpler. Raven was specifically designed

to separate the numerical algorithms from the physical process representations so that numerical routines could be modified with relative ease. This disentangling of numerical solvers and physical process representations is the main attribute that distinguishes Raven from existing distributed hydrological models. Raven possesses the unique ability to allow the modeller to change not only the physical process algorithms used to model a basin but also allows the user to change the numerical solver. By using Raven to test multiple numerical methods at multiple timesteps, operator splitting error impacts can be explored and be better understood. Raven's design, construction and operation is further described in chapter 3.

Several numerical methods have been added to Raven's library in order to increase the accuracy of distributed hydrological modelling while maintaining computational efficiency. These solvers have been used within Raven to simulate the water balance of the Nith River Basin in southwest Ontario, Canada. The solvers were first used to establish "mathematical truth" for the simulation. Via comparison with the truth, the performance of various numerical methods were assessed for accuracy and numerical error generated by numerical method choice. The improved numerics are shown to more accurately solve the mathematical problem statement (i.e., coupled ODEs) than the traditional numerical solvers used within distributed hydrological models.

1.3 Thesis Organization

This thesis is organized into several chapters that will explain the background of this research, the methods used to complete it as well as the results and discussion that highlight its value. The following sections provide a brief overview of each chapter in this thesis.

Chapter 1 Overview: Introduction

Chapter 1 provides an introduction to the research problem. The section describes the motivation behind this research, an overview of the research completed and an outline of the chapters within this thesis.

Chapter 2 Overview: Background

The research background in chapter 2 includes a comprehensive literature review, divided into two sections. The first section covers the current state of knowledge about distributed hydrological models and the methods used to solve them.

The second section discusses the mathematical background of current model methods and background on higher order methods that have been implemented within the software, Raven.

Chapter 3 Overview: Methods

Chapter 3 explains in detail how Raven was designed and implemented, as well as how the numerical solver routines were implemented into the distributed hydrological model. The final part of this chapter discusses the mass balance accounting that deals with error checking and the maintaining of mass within the system.

Chapter 4 Overview: Results and Discussion

This chapter describes the test site used during modelling tests and provides details of the model parameters used during numerical method testing. This chapter outlines the results of the modelling tests discussing the benefits and shortfalls of multiple numerical schemes when applied within distributed hydrological models. Further discussion focuses on the significance and impacts that the results have within the current state of research in this field.

Chapter 5 Overview: Conclusions

A conclusions chapter (Chapter 5) summarizes this research and highlights future directions that could be taken within this area of study.

Supplemental Sections Overview: Appendices and References

Appendices and a reference section are found at the end of this thesis.

Chapter 2

Background

In order to properly understand the impacts and significance of numerical method choice in hydrological models, it is necessary to first understand what hydrological models are and how they function. The following sections address what hydrological models are, the structures and capabilities of select existing hydrological models, the numerical methods used in those models and the introduction of additional numerical methods for use in distributed hydrological models.

2.1 Literature Review

2.1.1 Hydrological Modelling

Hydrological models are used to predict the movement of mass and/or energy through the hydrological cycle, depicted in figure 2.1. When modelling, the hydrological cycle is typically thought of in terms of storage compartments and hydrologic processes. Water and energy are stored in a set of storage compartments (e.g., soil, lakes, aquifers) and hydrologic processes (e.g., infiltration, runoff, percolation) move water and energy between these compartments.

Hydrological models attempt to provide a quantitative understanding of hydrologic regimes by establishing continuous mathematical relationships between the various components of the hydrological cycle [Crawford and Linsley, 1966]. In general, a numerical hydrological model divides up a watershed into smaller parts (e.g., grids, subbasins) and then, for each individual area, calculates the change in storage (e.g., water and/or energy) over time due to the influence of hydrological processes

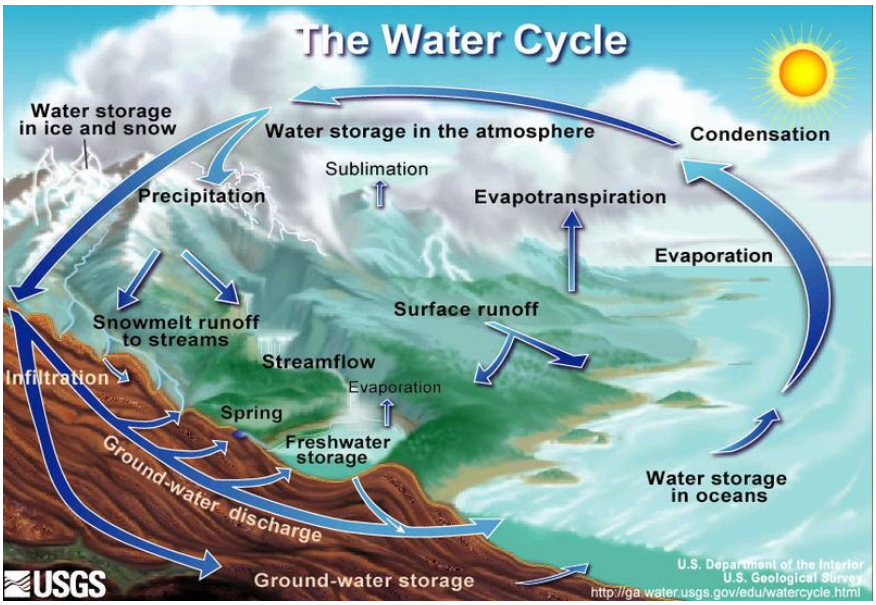

Figure 2.1: Hydrologic cycle showing hydrologic processes and storage units, from USGS [2008]

(e.g., evaporation, infiltration, etc.). Hydrological processes are defined in hydrological models by algorithms that mathematically represent how water or energy is redistributed over time. Each hydrological process is usually represented using a physically or conceptually based process algorithm [Dingman, 2002]. Physically-based process algorithms use equations that are derived from basic physics (i.e., conservation of mass/ energy, diffusion, etc.) and conceptually-based process algorithms use empirical or regression relationships established from field data (i.e., cases where a process is considered to be proportional to the amount of water/ energy stored in a particular storage unit) [Dingman, 2002]. These two types of process algorithms make up the majority of hydrological model components.

Many hydrological modelers fail to evaluate the fluxes that occur in the individual storage compartments and rely only upon total watershed outflow in order to justify the accuracy of the simulation. As a result, these models do not provide an understanding of what is happening with the processes involved within the watershed [Lee et al., 2007]. The failure to provide an understanding of process activity results in many models falling short of being robust, efficient and accurate for simulating distributed hydrological systems. A model needs to be able to simulate all aspects of its intended use and as models become more complex (and subsequently all encompassing), it becomes more difficult to maintain an accurate representation of the myriad of interactions within a watershed. In general, simple models (e.g.,

models representing fewer processes) are more effective at simulating their intended goal than complicated models (e.g., several processes or highly discretized models). However the use of simple models is limited to simple problems and situations [Borah et al., 2007]. It is often necessary to find a balance between a simple model and a complex model in order to have a robust model that is accurate and efficient [Borah and Bera, 2003].

Hydrological models left the world of manual calculations and entered the world of computer simulations in the 1950s and 1960s. Singh and Frevert [2006] reports that the first digital hydrological model available for distributed use was the Stanford Watershed Model IV . It was used to predict streamflows over short timescales provided enough meteorological data was available. Since then, models have continued to evolve and grow becoming more complex with increased capabilities. This means that more processes and more storage compartments are included in models, which necessitates an increase in the complexity of the algorithms that define the model.

2.1.2 Distributed Surface Water Modelling

Hydrologic systems are typically modelled using deterministic or stochastic models. Deterministic models use variables that have a single fixed value at any point in time whereas stochastic models use variables that are random and are described by probability distributions [Chow et al., 1988].

For the purposes of this thesis, only deterministic models will be reviewed. As previously mentioned, a deterministic model is one where the outcome of the model is determined through known relationships between state variables and their interactions. A deterministic model will always produce the same results given the same input data. Deterministic models can be classified into two categories; lumped and distributed [Chow et al., 1988]. Lumped models treat a watershed as a single system without lateral variability whereas distributed models are spatially discretized. Both types of model require temporal discretization, simulating the movement of water within a watershed over a period of time [Dingman, 2002]. However, because distributed hydrological models are discretized, they can simulate and predict the spatial variability of water and/or energy movement within a basin. They are capable of calculating watershed runoff and the amount of water routed through channels and rivers, like a lumped model, but can also be used to calculate other diagnostic variables, such as soil moisture.

Distributed models can be spatially discretized into grids, hydrological response units (HRUs) and/or grouped response units (GRUs). Each sub-basin of a watershed is composed of one or more hydrological response units that are used to characterize the area to be modelled. The grids and response units are used to represent unique combinations of land use, land cover and soil characteristics that are used to parameterize the model [Dingman, 2002]. Each grid's surface characteristics are described through the use of digital elevation maps (DEMs) or its slope, aspect and area. The sub-basins require spatially distributed meteorological data such as temperature, precipitation, wind speed and other necessary forcing functions (e.g., shortwave radiation, longwave radiation).

Distributed hydrological models are capable of modelling using different simulation durations in order to achieve the best representation of a system. Models are typically designed for either short-term forecasting (e.g., storm events) or long-term forecasting [Borah and Bera, 2003, 2004]. Short-term forecasting is beneficial for urban design and storm water planning. Long-term forecasting is necessary for water budget management and climate prediction. Both short-term and long-term simulations provide important information for the understanding of and interpreting of water resource data.

2.1.3 Numerical Methods for Distributed Surface Water Modelling

Existing distributed surface water models are typically composed of a number of mathematical submodels that may individually use a variety of numerical methods including numerical integration, root-finding and/or finite difference methods. Most of these submodels are designed based upon the principles of mass and energy conservation. Alternatively, these submodels may be 'simple' empirical models that are based on the regression of observation data.

Part of the difficulty with existing models is the inability to cater the submodels of an individual model to a complex problem. This is because the submodels of existing models have been designed with critical assumptions (e.g., that watersheds can be characterized by the combination of unconnected hydrological response units (HRUs) that are defined by land use and soil types) [Easton et al., 2008]. While this may work for locations with deep uniform soil profiles or locations where runoff is defined via infiltration-excess methods, this leads to models that are unable to

properly replicate various real world conditions (e.g., multiple connected HRUs, locations with shallow soils) and are therefore inaccurate for simulations [Easton et al., 2008].

Numerical Methods Used in Existing Models

Multiple numerical methods are employed in distributed hydrological models to solve and aggregate the influence of multiple process algorithms at every timestep. The traditional numerical method used to simulate hydrological systems is a form of operator splitting here called an "ordered series approach". Ordered series approaches use a fixed physical process order during a timestep. The redistribution of water and/or energy defined by the process algorithms are simulated in sequence and not treated as simultaneous like they occur in nature. Importantly, the ordered series is only at the global level; for each process submodel, finite difference schemes, finite element methods and/or analytical methods can be used.

As part of this research, a number of existing models and their numerical implementation were reviewed. The models surveyed represent a small sample of distributed hydrological models that are used to simulate surface water systems. A summary of these prototypical models follows. These models differ in discretization, process representation and numerical methods.

SWAT, a long term continuous model that was developed to predict water management, sediment and agricultural chemical yield impacts within watersheds [Borah and Bera, 2003, 2004] uses a numerical setup that includes several methods that solve for rainfall, overland runoff, channel runoff and subsurface flow. SWAT calculates the water balance through accounting daily or subdaily water budgets (e.g., SCS curve number, empirical equations) instead of mass conservation equations (e.g., kinematic wave, diffusive wave, dynamic wave) as can be found in many other models [Borah and Bera, 2003, 2004]. It uses a non-iterative ordered series approach and maintains a fixed timestep throughout simulations.

TOPMODEL, a conceptual semi-distributed topography-based model, breaks watersheds into grids or subwatershed units [Beven, 1997b]. TOPMODEL uses a raster grid of elevations in conjunction with its multiple direction flow algorithm to calculate the topography distribution of the watershed [Beven, 1997a]. A linear routing algorithm is used to calculate routing for the subwatershed [Beven, 1997a]. TOPMODEL uses an explicit Euler scheme to solve the model over time which can result in processes like baseflow increasing to a point where a basin can completely drain in one timestep [Beven, 1997b].

VIC, the Variable Infiltration Capacity model, is composed of process algorithms that are based on variable bucket concepts used in land surface schemes [Kavetski et al., 2003; Liang et al., 2003]. VIC considers both energy and water balances and is capable of representing multiple vegetation covers [Yuan et al., 2004]. VIC breaks down watersheds into subgrids and is capable of spatial variability with respect to precipitation and infiltration in order to calculate the water and energy balances [Yuan et al., 2004]. While not tied to any particular numerical method, Kavetski [2003] implemented the VIC model using an operator splitting method termed by Kavetski as 'the method of fractional steps' [Kavetski et al., 2003]. The operator splitting method used in VIC is similar in operation time to a Runge-Kutta second order method and therefore implies that the use of an explicit Euler method would half the runtime for VIC [Kavetski et al., 2003]. An implicit Euler method is used to solve for evapotranspiration and base flow and an analytical integration method for quickflow [Kavetski et al., 2003]. ARNO (a semi-distributed conceptual model named for the Arno river basin in Italy [Todini, 1988, 1996]) subsurface/ base flow runoff handles the slow flow calculations and an excess infiltration method is used for the quick flow runoff calculations [Liang et al., 2003].

WATFLOOD, a storm event model, is a combination of physically-based and conceptual equations that represent hydrological processes [Bingeman et al., 2006]. The processes that define vertical water movement are conceptually-based while the grid-to-grid routing equations are physically-based [Bingeman et al., 2006]. WATFLOOD does not use an advanced numerical method, its solver is based on the traditional ordered series method [Soulis, 2009]. WATFLOOD can have a variable timestep that is usually at a daily or subdaily interval. However, when WATFLOOD is run using a daily timestep, the model still performs calculations at hourly intervals [Kouwen et al., 2005].

CRHM, the Cold Regions Hydrological Model, was designed as a modular hydrological model with the intended use for modelling cold regions (e.g., northern Canada). The modular framework allows for the modification of process routines and the addition of new routines with ease [Pomeroy et al., 2007]. CRHM does not use advanced numerical methods to solve for individual subprocess results since most of the hydrological process modules are designed from physically-based process algorithms. This decreases the need for calibration but limits the model to processes that have already been physically modelled [Pomeroy et al., 2007]. CRHM uses an ordered series method to calculate hydrological model results. The order that CRHM calculates the redistribution rates of the processes is defined by the

order that the process modules are selected.

Lastly, FUSE, the Framework for Understanding Structural Errors, was designed to analyze the structural errors inherent within existing hydrological models [Clark et al., 2008]. Using VIC, TOPMODEL, SACREMENTO and PRMS for structural frameworks, FUSE is being used to analyze the impacts of model structural errors. FUSE uses the Newton-Raphson method, an implicit scheme with adaptive timesteps which requires iterative computations to achieve results within a specific convergence criteria [Clark et al., 2008]. The implicit scheme was deemed more accurate than the fixed-step ordered series explicit Euler method [Kavetski et al., 2003]. During the FUSE simulation, the operation order is fixed for the simulation duration.

Known Issues

The global numerical methods used in existing models are subject to numerical error that can arise due to insufficient timestep size, threshold behaviour and/or operator splitting. The following sections discuss these individual sources of numerical error in detail.

Timestepping Issues

Timesteps, the length of time that a model advances between calculations, have a large effect on the accuracy of a model. Timestep size in existing models may cause mathematical accuracy issues because the timestep increment is not an appropriate size to capture fine details (e.g., a model run with daily timesteps will not be able to represent storm events with high accuracy). Unfortunately, as timesteps shrink, computational runtime increases. For this reason, a short-term model is likely to be run using smaller timesteps than during a long-term model simulation. Short-term models use smaller timesteps because storm events are generally subdaily acivities with high variability in storage compartments. The smaller timesteps will produce more accurate results.

Multiple existing models (e.g. SWAT,FUSE) use only daily timesteps; though some are capable of using variable timesteps [Borah and Bera, 2003]. Models run using a daily timestep are prone to inaccuracies since the model is unable to replicate the sub-timestep details of the simulation. For example, a temperature-based process algorithm (e.g. snowmelt) may not accurately represent the process if a mean daily air temperature is used as opposed to a mean hourly air temperature. Daily averaged forcing functions may not represent the maximum and minimum values appropriately and therefore a process that may require the extremes will fail

to model well.

Any numerical approximation of a transient ODE requires time to be discretized into a number of timesteps. The finer the time discretization (i.e., smaller the timestep), the more accurate the approximation. The degree of accuracy depends on the size of the timestep used and the numerical method used to solve the model or submodel (e.g., ordered series, finite difference, etc).

A variety of models use different timestep sizes and have limitations in their timestep size flexibility and variability. SWAT performs well for simulating monthly flow values, but performs inadequately when used for daily results [Borah and Bera, 2004] because it uses a fixed timestep of one day [Borah and Bera, 2003]. Overall, this leads to the understanding that SWAT's timestep is too large to properly represent the intricacies of a watershed with a high degree of accuracy. Similar errors can be expected in other models with similar timestep requirements. TOPMODEL uses a variable timestep that can range from 1 to 24 hours [Beven, 1997b]. This provides the opportunity to increase the numerical accuracy of the model by reducing timestep size (especially around storm events). VIC can be used to model daily streamflow, monthly streamflow and predict future water resources [Yuan et al., 2004] using two timescales that can be used to calculate quickflow and slow flow runoff conditions [Liang et al., 2003]. This does not provide much flexibility or the ability to capture fine details. CRHM uses a variable timestep with no limit to the size of that timestep. FUSE was built to use daily timesteps which can produce numerical inaccuracies. Those inaccuracies may be reduced due to the use of higher order numerical methods (e.g., Newton-Raphson method), but time discretization errors are still an issue. An updated balance, reflecting current computational capabilities, needs to be found between timestep size and hydrological model runtime.

Threshold Issues and Stiff Ordinary Differential Equations

Threshold behaviour constrains many processes in surface water models [Kavetski et al., 2006]. These thresholds, where state variables or fluxes are constrained to a limited range of values, limit the flow and transport of mass and energy within the system. They act as switches, turning processes on and off as conditions are met (e.g., turning off infiltration when maximum soil moisture capacity is reached). A simple example of a threshold constraint is the degree day snowmelt model. In the real world, the rate of snowmelt increases gradually as the threshold point (e.g., melt temperature) is approached (Blue line on figure 2.2). However, within numer-

ical models, this transition is treated as instantaneous (Red line on figure 2.2). The presence of thresholds affects mathematical accuracy because the conditions that dictate when the processes turn on or off are often sensitive to the current solution, which is, in turn, sensitive to the numerical method [Kavetski and Kuczera, 2007]. These threshold constraints, while typically physically-based, can produce numerical problems when trying to model a basin as they are extremely non-linear. Non-linearity reduces the validity of error bound estimates for ordered series and Runge-Kutta methods, which assume continuous, threshold-free differential equations. In fact, due to the presence of thresholds, surface water models may be considered infinitely non-linear. The risk of over-draining, over-filling or having spontaneous behavioral switches within processes will generate complications that can produce errors in the model results. All surface water models contain these threshold constraints and no methods have yet been produced to fully correct for them.

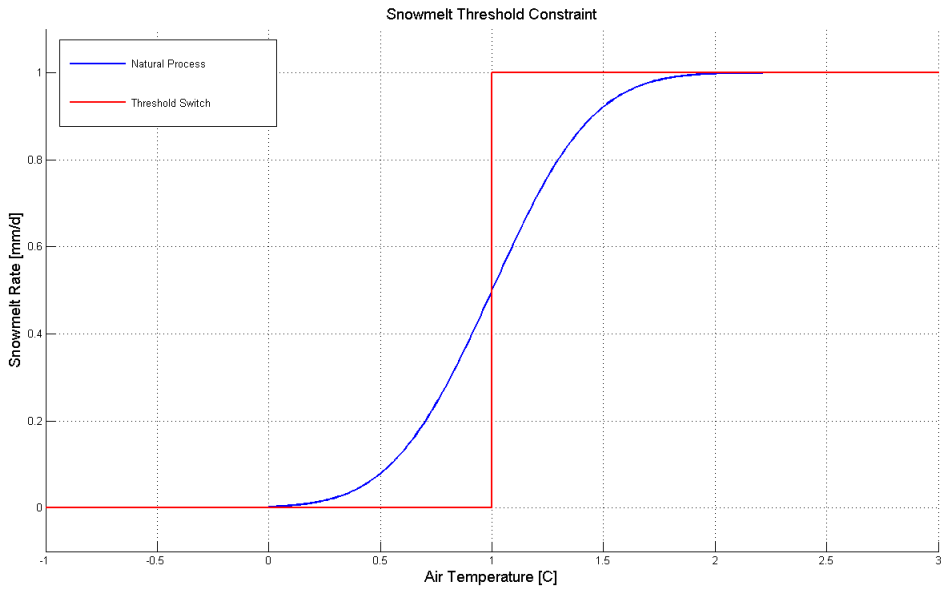

Figure 2.2: Simple example of threshold behaviour in models

A closely related problem is the issue of stiffness of the system of ODEs defining a hydrological model. A set of ODEs is considered stiff when the timestep size needed for stability is much smaller than the timestep size needed for accuracy [Hoffman, 1992]. This implies that timestep size needs to be reduced beyond the point needed to achieve mathematical accuracy in order to maintain the stability of the solution. Other criteria for stiff ODEs include when the timestep needed for

stability is so small that significant round-off errors are produced [Hoffman, 1992]. This occurs in surface water models often due to the use of coupled ODEs that each would independently require multiple timestep sizes. For stability, in order to solve the coupled ODEs with a high degree of accuracy, the timestep size used for a simulation needs to reflect the smallest timestep size in the coupled ODEs. Using the smallest timestep will increase the running time of a simulation and is therefore often undesirable. This leads to the use of inappropriate timestep sizes in models which can potentially produce large inaccuracies in results. The use of higher order numerical methods can help to overcome some of the problems inherent in stiff ODEs.

Model	**Numerical Method**	**Timestep**	**Thresholds**
SWAT	ordered series	daily	yes
TOPMODEL	explicit Euler	variable (1-24hr)	yes
VIC	ordered series	2 timescales	yes
WATFLOOD	ordered series	variable	yes
CRHM	ordered series	variable (any)	yes
FUSE	Newton-Raphson	daily	yes

Table 2.1: Summary of numerical methods in existing models

Operator Splitting Issues

Operator splitting may have a significant impact on the accuracy of models (see section 2.2.1) and is used in all models surveyed in table 2.1. As previously stated, the ordered series approach treats the set of coupled ODEs that define the hydrological model as an ordered list of processes to be solved successively in time. As each physical process rate is calculated, mass and/or energy is redistributed between storage units. At every timestep, the ordered series approach may produce errors as the stored water/ energy of the system are redistributed before the next physical process rate is calculated. These errors may be compounded from previous timesteps and may theoretically result in a significant impact on model accuracy by the end of the simulation. This introduces concerns regarding how model results can be used confidently for resource management when accuracy is in question.

2.2 Mathematical Background

Hydrologic models are defined by the algorithms and equations which they use to describe the redistribution of water and energy in the natural environment [Zhang

et al., 2008]. In order to solve the defining set of mixed ODEs/PDEs, the problem is typically discretized in time and space and solved using a number of mixed numerical methods. Most numerical methods for distributed surface water models (with the known exception of FUSE) use the ordered series approach as the global solver (i.e., the method used to connect the individual processes)(see table 2.1). The local solver (i.e., used to simulate a single process) might also use ordered series (e.g., most snow balance models) or might use a finite difference approach (e.g., soil water redistribution by solving Richards equation in CLASS [Verseghy, 1991]).

2.2.1 Operator Splitting and Operator Splitting Errors

Operator splitting is a popular technique used to solve complex problems where the original mathematical problem is split into a sequence of more manageable problems [Faragó, 2008]. Time-splitting is used to uncouple and solve the governing equations of a system often relating to transport/reaction, advection/dispersion or another practical problem involving energy balance and/or mass balance [Kanney et al., 2003]. The main goal in operator splitting is to replace the original model with one consisting of groups of sub-processes that can then be solved successively (rather than simultaneously) in time, allowing a solution to be obtained from several 'simpler' systems instead the one large one [Faragó, 2006]. However, in order to properly model the complex physical world, it is necessary to accurately model the effects of several different processes all acting concurrently [Faragó and Havasi, 2001] and understand the errors associated with those methods. At a small enough timestep, the successive and concurrent solutions should be identical.

Equations 2.1 - 2.5 present the general concept of how operator splitting approaches are applied within a model. For illustration, the following lone ODE is used:

$$\frac{\partial \phi}{\partial t} = M_1 + M_2 \tag{2.1}$$

where ϕ represents a storage unit $[L]$, t is time $[T]$, M_1 and M_2 are the influence of hydrological processes $[LT^{-1}]$.

This simple example can be used to explain how operator splitting is implemented in hydrological models. The ordered series method solves the above equa-

tion as follows:

$$\phi_*^{n+1} = \Delta t M_1\left(\phi^n\right) + \phi^n \tag{2.2}$$

$$\phi^{n+1} = \Delta t M_2\left(\phi_*^{n+1}\right) + \phi_*^{n+1} \tag{2.3}$$

i.e., first process M_1 modifies the storage, then M_2 uses the results of that calculation to obtain ϕ^{n+1}.

In contrast, the Euler method can be expressed as:

$$\phi_*^{n+1} = \Delta t M_1\left(\phi^n\right) + \phi^n \tag{2.4}$$

$$\phi^{n+1} = \Delta t M_2\left(\phi^n\right) + \phi_*^{n+1} \tag{2.5}$$

i.e., the process flux M_2 uses delayed information from the start of the timestep. Both methods separate the information of process M_1 and M_2.

The primary difference between these two applications (both forms of operator splitting) is that the Euler method does not use the most recent storage unit value to calculate the change in the storage unit. The ordered series approach continuously modifies storage unit values over the course of the timestep, using the most recent information, whereas the Euler method calculates the redistribution rates for all simulated processes using the storage values from the beginning of the timestep and only applies the changes to the storage units afterwards.

The primary advantage of OS methods is that different numerical schemes can be used to solve the subcomponents of a model at each stage due to the uncoupling of the system's governing equations [Valocchi and Malmstead, 1992]. This is why OS is used almost exclusively in surface water models, which are always built from component submodels. More efficient or appropriate solvers can be applied to the individual elements of the system than can be applied to the fully coupled algorithms [Kanney et al., 2003]. Other advantages that should be mentioned include the easy implementation of OS when different modules or codes are coupled, it is possible to use multiple numerical methods to solve the uncoupled system and codes can be readily modified for parallel computation [Jacques et al., 2006].

Even though the use of operator splitting methods is ubiquitous, no studies have been found that report the effects of different operator splitting methods within surface water models. However, several studies have looked at the impacts of OS in subsurface water modelling. Methods for minimizing operator splitting errors used

by Kanney and Carryrou [Kanney et al., 2003; Carrayrou et al., 2004] have been applied to advection/ dispersion problems but are not appropriate for minimizing errors in surface water models. Typically, non-iterative OS methods are used when designing numerics and algorithms [Simpson and Landman, 2008]. However, OS errors can theoretically be reduced if iterative methods are used [Kanney et al., 2003]. The use of high order iterative numerical schemes can reduce the operator splitting errors and solve the system of coupled ordinary differential equations as if it was a simultaneous operation.

OS methods are applied exclusively in surface water models. In fact, many hydrological model developers feel that a specific order of hydrological processes is 'correct' [Pomeroy, 2008]. Nature performs all hydrological process operations concurrently, so any 'order of operation' specified by a user is not representative of the actual environment. Even with only a few hydrological processes, the operation order can vary in multiple ways. These variations can lead to large differences in the end results from the model.

2.2.2 Runge-Kutta Methods

In most fields, coupled ODEs are most commonly solved using Runge-Kutta methods. Unlike the ordered series approach, when using Runge-Kutta methods, the order in which the processes are calculated does not matter. The Runge-Kutta methods calculate an approximation to the solution for the entire system of ODEs before applying that solution. This approximation is independent of the operation order of the solution. This can replicate the 'simultaneous' solution of the system. In this thesis, both the explicit Euler and implicit iterative Heun methods are tested and evaluated to see their impact on hydrological model results.

Runge-Kutta methods are based on rearrangement of the Taylor series expansion of the dependent variable. For the Euler method, which is a 1^{st} order method, only terms from the Taylor series expansion that are 1^{st} order are included in the calculation of the solution. Likewise, the Heun method, a 2^{nd} order method, only uses terms from the Taylor series expansion that are 2^{nd} order and below [Hoffman, 1992].

All Runge-Kutta methods are used to solve an initial value problem, here ex-

pressed as a single ODE:

$$\frac{\partial \phi}{\partial t} = f(t, \phi), \qquad \phi(0) = \phi_0 \tag{2.6}$$

where $\frac{\partial \phi}{\partial t}$ is the finite difference approximation of the solution, $f(t, \phi)$ is a function of time and an independent variable, ϕ is the independent variable of the initial value problem and ϕ_0 is initial value of the independent variable.

The explicit Euler Method takes the form:

$$\phi_{n+1} = \phi_n + \Delta t f_n \left(t_n, \phi_n\right) \tag{2.7}$$

where ϕ_{n+1} is the solution to the problem for the current timestep, ϕ_n is the solution from the previous timestep, Δt is timestep size, f_n is a function of t and ϕ from the previous timestep and t_n is the time at the previous timestep. The solution and variables from previous timesteps are equal to the solution and variable values at the start of the current timestep.

The explicit Euler method makes a prediction of the solution to the ODE by using information from only the beginning of the timestep. It uses the initial values of the current timestep in an attempt to approximate the exact solution [Hoffman, 1992]. This approach is a 1^{st} order finite difference scheme which is prone to truncation errors that create inaccuracies in results.

The Heuns method, a 2^{nd} order numerical scheme, takes the form:

$$\phi^P_{n+1} = \phi_n + \Delta t f_n \left(t_n, \phi_n\right) \tag{2.8}$$

$$\phi^C_{n+1} = \phi_n + \frac{1}{2}\Delta t \left(f_n + f^P_{n+1}\right) \tag{2.9}$$

where ϕ^P_{n+1} is the prediction of the solution for the current timestep, ϕ^C_{n+1} is a corrector step for the current timestep and f^P_{n+1} is the simple ODE function using t and ϕ from the prediction step.

The Heuns method uses an explicit Euler scheme to make a prediction of the solution and an implicit method to correct that prediction. The equations above show the predictor (equation 2.8) and the corrector (equation 2.9). This method uses information from the beginning of the timestep to make the initial prediction and then attempts to correct that prediction using information from a combination of the initial information and the predicted information [Hoffman, 1992]. This approximation is more accurate than the Euler method, however it does take more

computational runtime.

An iterative Heuns method can also be used to solve ODEs and takes the form:

$$\begin{aligned}
\phi_{n+1}^{0} &= \phi_n + \Delta t f\left(t_n, \phi_n\right) \\
\phi_{n+1}^{1} &= \phi_n + \frac{1}{2}\Delta t\left(f\left(t_n, \phi_n\right) + f\left(t_{n+1}, \phi_{n+1}^{0}\right)\right) \\
\phi_{n+1}^{2} &= \phi_n + \frac{1}{2}\Delta t\left(f\left(t_n, \phi_n\right) + f\left(t_{n+1}, \phi_{n+1}^{1}\right)\right) \\
&\vdots \\
\phi_{n+1}^{k} &= \phi_n + \frac{1}{2}\Delta t\left(f\left(t_n, \phi_n\right) + f\left(t_{n+1}, \phi_{n+1}^{k}\right)\right)
\end{aligned} \tag{2.10}$$

The iterative Heuns method performs the same initial two steps that are done to solve the non-iterative Heuns method. In order to produce a more accurate approximation of the exact solution, the Heuns method can use further corrector steps in an attempt to converge the solution to a 'best' approximation. After each corrector step, the solution approximation is checked against the user specified convergence criteria to see if convergence (as defined by the user) has been reached. This numerical scheme is more accurate than the non-iterative Heuns method and Euler method, but is also more complicated to implement, particularly for a system of ODEs. Equations and an extension to the solution of multiple ODEs using the discussed numerical methods is addressed in chapter 3.

2.3 Background Synopsis

As has been discussed, there is a lack of studies into the significance and impacts of operator splitting errors in distributed surface water models. The author was unable to find any literature addressing operator splitting issues with regards to distributed surface water models. However, it is clear from related literature that the choice of numerical method has a large impact upon the results of solved coupled ODEs. It would be beneficial for additional studies to be done to provide detailed analyses into how detrimental operator splitting errors are with regards to other aspects of hydrological modelling such as climate/ carbon modelling and contaminant transport modelling. Other studies that would be helpful include research into the effects of higher order numerical methods and their implementation for these hydrological model extensions.

Chapter 3

Methods

The following section discusses the methods, software design and algorithms used to complete the objectives of this thesis: (1) to understand and quantify operator splitting errors within hydrological models and (2) to test better approaches for numerical implementation of hydrological models. Section 3.1 describes the design and functions of the object-oriented software package Raven. The two sections (3.2.1 & 3.2.2) that follow discuss the implementation of ordered series methods and Runge-Kutta methods within Raven. The final section shows how the mass balance budget is maintained and verified.

3.1 Object Orientation

3.1.1 Motivation

A new hydrological model, Raven [Craig and Snowdon, 2010], has been designed and built to understand the impacts of operator splitting (OS) and to test numerical methods in distributed hydrological models. It was uniquely designed to maintain a separation between physical process representations and numerical solvers. This design choice eases the introduction and testing of a variety of numerical solution methods for use in distributed hydrological models. Rather than use an existing hydrological model, which does not have multiple numerical method solvers, Raven has been designed and built with the ability to add and use multiple numerical solvers. The use of multiple numerical methods is able to be implemented with little to no change required to the bulk of the model software. Raven's class structure, parsing libraries, initial numerical solvers (e.g., ordered series, Euler) and the

original input files were designed and built by James R. Craig [Craig and Snowdon, 2010]. The author, throughout this research, added to the design and built new physical process routines, new numerical solvers (e.g., Heun, iterative Heun) and new data structures as were needed to emulate existing hydrological models, including VIC and TOPMODEL.

Raven was designed and built through the use of object-oriented programming. Object-oriented programming focuses on using discrete, reusable sections of code, built into modules (objects) that are capable of manipulating their own data structure [Booch, 1994]. A class defines the abstract characteristics of a category (e.g., Class River might contain depth, watershed area, etc.). An object is a particular instance of a class (e.g., Mackenzie could be an object of the River class).

Using object-oriented programming in the C++ language, Raven was designed using multiple classes and objects to represent the various elements within the model, such as rivers, soils, etc. By creating these objects and classes, Raven is a robust model with extensive flexibility. This allows Raven to be adapted with ease to suit the needs of the model and modeller.

3.1.2 Distributed Surface Water Modelling Library: Raven

3.1.2.1 Overview

Distributed hydrological models make a prediction of mass/ energy values stored within one or more storage units (e.g., snow, soil, surface water, etc.). To calculate the mass/ energy balance within these storage units, it is necessary to divide up a watershed into multiple homogenous hydrological response units (HRUs). The HRUs generally share similar characteristics including land type, land use, soil profile and vegetation cover. In each HRU, multiple storage units exist and can be thought of as buckets containing the water or energy that is present in a particular location within a hydrological response unit. Hydrologic processes act to redistribute the water and energy between the storage units. The redistribution is mathematically represented as coupled sets of ODEs and PDEs for each HRU. After the ODEs and PDEs for the HRUs are defined, storage within an HRU can be solved using various numerical methods that vary in their computational accuracy and efficiency. Each HRU can have its mass routed to and through channels that connect the multiple HRUs and allow hydrographs to be created that describe channel flow over time.

Most deterministic hydrological models are based upon a system of ordinary differential equations that are coupled and serve to define the mass/ energy balance in the storage units of an HRU. The general system of mass/ energy balance equations can be described mathematically with:

$$\frac{d\phi_i}{dt} = \sum_{j=1}^{NS} M_{ij}(\{\phi\},\{P\}) + Q_i(\{\phi\},\{P\}) \quad \text{for } i\text{=1 to } NS \tag{3.1}$$

where ϕ_i is the i^{th} water/ energy storage variable [L^3 or EL^2], M_{ij} is the combined rate of one or more hydrological processes that moves water or energy from the jth storage unit to the ith storage unit [LT^{-1}] which is a function of some subset of the vector of state variables, $\{\phi\}$ and the physical and empirical parameters, $\{P\}$ that define the HRU. Q_i [LT^{-1}] is an external flux to or from the HRU (typically precipitation or surface water gains/losses). NS represents the number of state variables that are being simulated in a model run [-]. All storage values are divided by the watershed area.

Specific mass balance equations are used within Raven to describe the flow of mass and energy to and from the various storage units. These mass/ energy balance equations are defined by the user to dictate which storage units receive and lose mass and energy at rates defined by the physical process algorithms. These 'rates of change', M_{ij}, are calculated by physical process algorithms and determine the net amount of water and/or energy that is transferred from or to a storage compartment during a timestep. Multiple mass/ energy balance equations can be formulated at one time and each one will define the connections that exist between individual state variables. As an example of how these systems of ODEs are structured, equation 3.2 and equation 3.3 are used by Clark et al.[2008] to define the mass balance within FUSE when configured to replicate the variable infiltration capacity (VIC) model for the upper soil layer and lower soil layer respectively .

$$\frac{\partial S_1}{\partial t} = (P_p - q_{sx}) - e_1 - q_p - q_{if} - q_{ufof} \tag{3.2}$$

$$\frac{\partial S_2}{\partial t} = q_p - e_2 - q_b - q_{sfof} \tag{3.3}$$

where S_1 is the upper soil layer [L], S_2 is the lower soil layer [L], P_p is precipitation [LT^{-1}], q_{sx} is surface runoff [LT^{-1}], e_1 is upper soil evaporation [LT^{-1}], q_p is precolation from upper to lower soil layer [LT^{-1}], q_{if} is interflow [LT^{-1}], q_{ufof} is overflow from free storage in the upper soil layer [LT^{-1}], e_2 is evaporation from the lower

soil layer $[LT^{-1}]$, q_b is baseflow $[LT^{-1}]$ and q_{sfof} is overflow from free storage in the lower soil layer $[LT^{-1}]$. Equations 3.2 and 3.3 are specific mass balance equations that follow from the general mass balance equation (equation 3.1). S_1 and S_2 correspond to ϕ, a storage unit where water or energy amounts are temporarily stored. P_p corresponds to Q_i, the external forcing input of precipitation. Variables q and e are physical process algorithms that corresponds to M_{ij}, the rates of hydrological fluxes.

Mass balance equations similar to equations 3.2 and 3.3 can be replicated in Raven to define all aspects of a basin's mass/ energy flow between state variables. They can be constructed to reflect any dynamic that is desired and this adds to Raven's high flexibility and robustness.

3.1.2.2 Raven: Operation

The pathway of a simulation within Raven is shown in figure 3.1. When the simulation begins, Raven parses four input files that define the characteristics of the simulation. The main input file (*.rvi) contains information detailing the model start date, the simulation duration, identifies the solver method, the names and number of state variables and the physical processes with their connections that will be used during the simulation. The HRU properties file (*.rvh) provides information about the river/channel class, the HRU properties (e.g., index, elevation, area, location, land use, vegetation cover, soil profile, slope and aspect) and defines the initial conditions for all state variables. The timeseries property file (*.rvt) contains all meteorological gauge information. It provides the number of and name of all gauges as well as the daily rain, snow, minimum temperature and maximum temperature values for the model duration. The Raven class property file (*.rvp) defines the soil profiles (e.g., percent sand, clay, organic soil breakdown), all soil characteristics (e.g., hydraulic conductivity, thermal properties, etc.), vegetation classes (e.g., land cover type, vegetation properties), seasonal vegetation properties and aquifer properties.

After the input files have been parsed, the model is initialized. At this point, tracking of the mass/ energy balance is started, the timeseries for the meteorological gauge data is loaded and the basins, subbasins, HRUs and physical processes are defined and initial conditions are set. This information is used in combination to set the initial mass/ energy balance values. The timestepping portion of the simulation begins here. The initial conditions are used to update all forcing functions (e.g.,

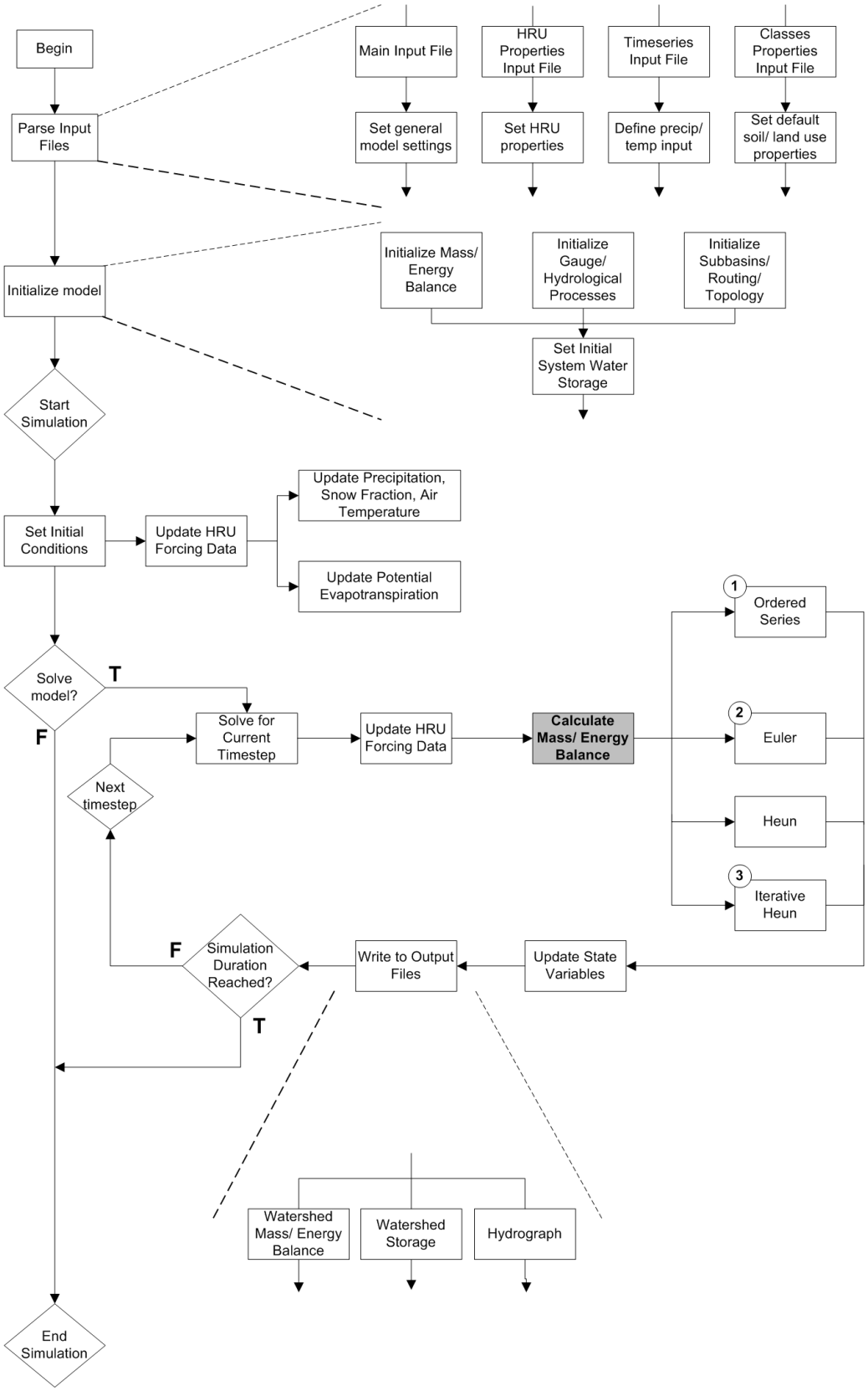

Figure 3.1: Flowchart of Raven's general program flow. The specifics of (1),(2) and (3) are shown in figure 3.2, figure 3.3 and figure 3.4

precipitation values, temperature values, etc. for the current timestep) and to define the potential evapotranspiration for the first timestep. If no errors from the input files or forcing functions have been flagged, the simulation is ready to progress forward through the timesteps until completion.

The simulation progresses by marching forward in time, solving the mass/ energy balance for each discrete timestep. To solve the water/ energy balance over the current timestep, Raven first updates the forcing functions (e.g., precipitation values, air temperature values, etc.) and then proceeds to use the numerical solution method as specified in the main input file to calculate the 'rates of change' (M_{ij}) for each physical process (e.g., infiltration, runoff, canopy drip, etc.). Mass and/or energy is moved from and to the storage units (e.g., soil, canopy, depression) based on the rates of change as determined by the process algorithm. When the conditions of the numerical solver have been satisfied for the current timestep, the output files are updated. The process is repeated for each timestep until completion.

3.1.2.3 Hydrologic Processes

Raven has an extensive library of hydrological process algorithms (i.e., equations representing real world processes such as evaporation, sublimation or infiltration). In the real world, each hydrologic process is governed by a complex set of physical laws which are often too difficult to model explicitly. Instead, simplified physical or empirical relationships are commonly used to estimate net water and energy fluxes due to hydrological processes at the watershed scale. Since they are empirical, there is no agreed upon way to calculate fluxes and different, equally valid, representations may be available for each process [Beven, 2001].

The full list of physical processes incorporated into Raven at this point has been tabulated for reference. Table A.1 details the evaporation and canopy processes, table A.2 details the soil related processes, table A.3 details the snow related processes and table A.4 shows the overland processes. Since only a handful of these processes have been used during the testing and modelling during this research, only those processes will be explained in detail in the following subsections.

Surface Runoff

Surface runoff has been adapted from FUSE [Clark et al., 2008] using the VIC runoff equation [Wood et al., 1992]. It uses a saturation-excess mechanism which determines how much precipitation will be runoff based upon the saturated area

that the precipitation falls on. The saturated area for the VIC configuration is determined using the following:

$$A_{sat} = 1 - \left(1 - \frac{S_1}{S_{1,\max}}\right)^b \tag{3.4}$$

where A_{sat} represents saturated area[-], S_1 represents the upper soil layer moisture $[L]$, $S_{1,\max}$ represents the maximum storage of the upper soil layer $[L]$, and b is the VIC exponent [-]. Using the saturated area, it is possible to calculate the surface runoff using:

$$\frac{\partial S}{\partial t}\bigg|_{Run} = A_C P_p \tag{3.5}$$

where $\frac{\partial S}{\partial t}\Big|_{Run}$ represents surface runoff $[LT^{-1}]$ and P_p is precipitation $[LT^{-1}]$.

Snowmelt

Snowmelt is highly variable depending on time of year, temperature, solar radiation and land cover [Dingman, 2002]. The Hybrid Snowmelt method is a basic snowmelt process that shows higher prediction accuracy then the Degree Day Snowmelt method but is not complex enough to incorporate the full energy fluxes that exist within the snowpack [Dingman, 2002]. Developed by Kustas et al. [1994] and Brubaker et al. [1996], daily snowmelt can be calculated using:

$$\frac{\partial S}{\partial t}\bigg|_{Melt} = \left(\frac{K + L}{\rho_w \lambda_f}\right) + M_r T_a \tag{3.6}$$

where $\frac{\partial S}{\partial t}\Big|_{Melt}$ is the rate of snowmelt $[LT^{-1}]$, K is shortwave radiation$[EL^{-2}T^{-1}]$, L is longwave radiation$[EL^{-2}T^{-1}]$, ρ_w is the density of water$[ML^{-3}]$, λ_f is the latent heat of fusion (334 $KJKg^{-1}$), T_α is the temperature of air$[\theta]$ and M_r is the restricted melt factor $[LT^{-1}\theta^{-1}]$. The restricted melt factor can be derived from basic equations but has been shown to be approximately 2 mm day^{-1} °C.

Percolation

The percolation routine has been adapted from the FUSE [Clark et al., 2008] implementation of the variable infiltration capacity (VIC) model for percolation [Liang et al., 1994; Demaria et al., 2007]. This method is used to represent vertical transport of water from the upper soil layer to the lower soil layer as well as vertical transport of water from the lower soil layer to deep groundwater storage (i.e.,

aquifer). The VIC configuration for percolation within FUSE is defined as:

$$\left.\frac{\partial S}{\partial t}\right|_{Perc} = k_u \left(\frac{S_1}{S_{1,\max}}\right)^c \tag{3.7}$$

where $\left.\frac{\partial S}{\partial t}\right|_{Perc}$ is the percolation rate $[LT^{-1}]$, k_u is the potential percolation rate $[LT^{-1}]$, S_1 is the water content of the upper soil layer $[L]$, $S_{1,\max}$ is the maximum water content of the upper soil layer $[L]$ and c is the percolation exponent [-]. This configuration for percolation is essentially equal to the gravity drainage term in Richards equation [Clark et al., 2008].

Baseflow

The baseflow algorithm for testing Raven was adapted from the FUSE [Clark et al., 2008] implementation of the variable infiltration capacity (VIC) model for percolation [Todini, 1996; Liang et al., 1996]. This representation allows horizontal transport of water from the lower soil layer to surface water storage. It is a nonlinear storage function that is used in conjunction with a single storage unit of infinite size [Clark et al., 2008] and is defined by:

$$\left.\frac{\partial S}{\partial t}\right|_{Base} = k_s \left(\frac{S_2}{S_{2,\max}}\right)^n \tag{3.8}$$

where $\left.\frac{\partial S}{\partial t}\right|_{Base}$ is the baseflow rate $[LT^{-1}]$, k_S is the potential baseflow rate $[LT^{-1}]$, S_2 is the water content in the lower soil layer $[L]$, $S_{2,\max}$ is the maximum water content of the lower soil layer $[L]$ and n is the baseflow exponent [-].

Evaporation Routines

The Potential Evaportranspiration (PET) method used for open water evaporation calculation is the Penman-Combination equation [Dingman, 2002].

$$PET = \frac{m_{et} \cdot (K + L) + \gamma \cdot e_v \cdot \rho_w \cdot \lambda_v \cdot v \cdot e_{sat} \cdot (1 - W_a)}{\rho_w \cdot \lambda_v \cdot (m_{et} + \gamma)} \tag{3.9}$$

Open Water evaporation rates are calculated as a simple fraction of PET. It is used as a constant percentage of the PET value for that timestep.

$$E_{ow} = PET \cdot F_{PET} \tag{3.10}$$

PET for vegetated surfaces is calculated using the Penman-Monteith method

[Dingman, 2002]. The Penman-Monteith equation makes use of the Penman-Combination model [Dingman, 2002], but is modified to include atmospheric conductance. This allows the Penman-Monteith equation to be used for a vegetated surface by including canopy conductance. This equation has become the most common approach for estimating evapotranspiration.

$$PET = \frac{m_{et} \cdot (K + L) + \rho_a \cdot c_a \cdot C_{at} \cdot e_{sat} \cdot (1 - W_a)}{\rho_w \cdot \lambda_v \cdot [m_{et} + \gamma \cdot (1 + \frac{C_{at}}{C_{can}}]} \tag{3.11}$$

where m_{et} is the slope of the relation between saturated vapour pressure and temperature $[P\theta^{-1}]$, ρ_a is the density of air $[ML^{-3}]$, c_a is the heat capacity of air (1.00 x $10^{-3} MJkg^{-1}K^{-1}$), C_{at} is atmospheric conductance $[LT^{-1}]$, e_{sat} is saturated vapour pressure $[P]$, W_a is relative humidity [-], ρ_w is the density of water $[ML^{-3}]$, λ_v is the latent heat of vaporization (2260 $KJkg^{-1}$), γ is the psychrometric constant (0.066 $kPaK^{-1}$) and C_{can} is the canopy conductance $[LT^{-1}]$.

To calculate soil evaporation, the sequential soil evaporation routine has been incorporated from FUSE [Clark et al., 2008]. This method can be used for a VIC [Clark et al., 2008] configuration or to replicate TOPMODEL [Clark et al., 2008]. To replicate TOPMODEL, evaporation from the lower soil layer is ignored. The sequential soil evaporation method uses the PET calculated for the current timestep and the upper soil layer attempts to satisfy the PET demand. If the upper soil layer doesn't satisfy the demand, the remaining demand is met by the lower soil layer. Sequential soil evaporation is defined by:

$$\left.\frac{\partial S}{\partial t}\right|_{SE_1} = PET \frac{\min(S_1^T, S_{1,\max}^T)}{S_{1,\max}^T} \tag{3.12}$$

$$\left.\frac{\partial S}{\partial t}\right|_{SE_2} = (PET - e_1) \frac{\min(S_2^T, S_{2,\max}^T)}{S_{2,\max}^T} \tag{3.13}$$

where $\left.\frac{\partial S}{\partial t}\right|_{SE_1}$ and $\left.\frac{\partial S}{\partial t}\right|_{SE_2}$ are the evaporation rates from the upper and low soil layers respectively $[LT^{-1}]$, S_1^T and S_2^T are the tension water content of the upper and lower soil layers respectively $[L]$, $S_{1,\max}^T$ and $S_{2,\max}^T$ are the maximum tension storage in the upper and lower soil layers respectively $[L]$.

3.2 Solver Implementation

There are four numerical solvers that have been implemented within Raven. They include the traditional ordered series approach and three Runge-Kutta methods (Euler, Heun and iterative Heun) that have not previously been used in surface water models.

3.2.1 Ordered Series Implementation

3.2.1.1 Overview

The most common and straightforward method used within distributed hydrological models is the ordered series approach. This method solves based on the operation order. The advantages of this method include minimal computational burden and an exact mass balance that occurs since water/ energy are moved between storage units in sequential steps. There is also no need to worry about the interacting processes in software code. The disadvantages are that as each process rate of change is applied, potentially artificial deficits and surpluses are created in storage units that will alter how much mass/ energy is available for the next process to access and manipulate.

The ordered series method calculates the influences of physical processes in the order specified by the user. For example, if the order specified is infiltration, percolation, evaporation, then the method will simulate infiltration first, then percolation (using the storage compartment values after infiltration has occurred) and finally evaporation (using the storage quantities after both infiltration and percolation have occurred). The fact that the processes are calculated in a set order may lead to accuracy issues by moving inappropriate quantities of mass and/or energy to and from storage units. For example, in dry regions, percolation may "remove" all of the water from the soil before evaporation (the actual dominant process) is allowed to act on the soil. This may cause incorrect amounts of mass and energy to be moved in subsequent calculations. This act of competing processes may result in compounded inaccuracies over time.

The ordered series method has been included in Raven as a basis of comparison. It allows for examination of the impact and significance of operator splitting errors on distributed hydrological model results. Its secondary use is as a surrogate for the

traditional solution method and to enable comparison with the newly implemented numerical solvers.

3.2.1.2 Implementation Strategy

The algorithmic implementation of the ordered series method is shown in Figure 3.2. From the main body of Raven, the ordered series solver method is called to calculate the rates of change in storage due to each process. This method moves water and energy, at the rate calculated, from the storage units in the order specified by the user. Every process, including precipitation, is subject to the order. In general, precipitation for the timestep is added at the beginning of the timestep, but this convention can also be changed if the simulation requires it to be so. The ordered series solver applies the process 'rates of change' to their associated storage units for each timestep and for each HRU. The ordered series approach may be expressed mathematically as:

$$\begin{aligned} \{\phi\}^{*} &= \{\phi\}^{t} + \Delta t \left([M]^{1,t} + \{Q\}^{1,t} \right) && (3.14) \\ \{\phi\}^{*} &= \{\phi\}^{*} + \Delta t \left([M]^{2,t} + \{Q\}^{2,t} \right) && (3.15) \\ &\vdots \\ \{\phi\}^{t+\Delta t} &= \{\phi\}^{*} + \Delta t \left([M]^{NP,t} + \{Q\}^{NP,t} \right) && (3.16) \end{aligned}$$

where NP is the number of processes being simulated and $[M]^{NP,t}$ uses only the most recent state variable vector, ϕ^{*}. The rest of the notation follows that of equation 3.1.

3.2.1.3 Computational Considerations

The ordered series approach is the least computationally intense of algorithms considered here and does not require as much computational speed or memory during simulations when compared to the other algorithms being considered.

3.2.2 Runge-Kutta Implementation

3.2.2.1 Overview

The Runge-Kutta solvers implemented within Raven include an explicit Euler method, an implicit Heun method and an iterative implicit Heun method. A disad-

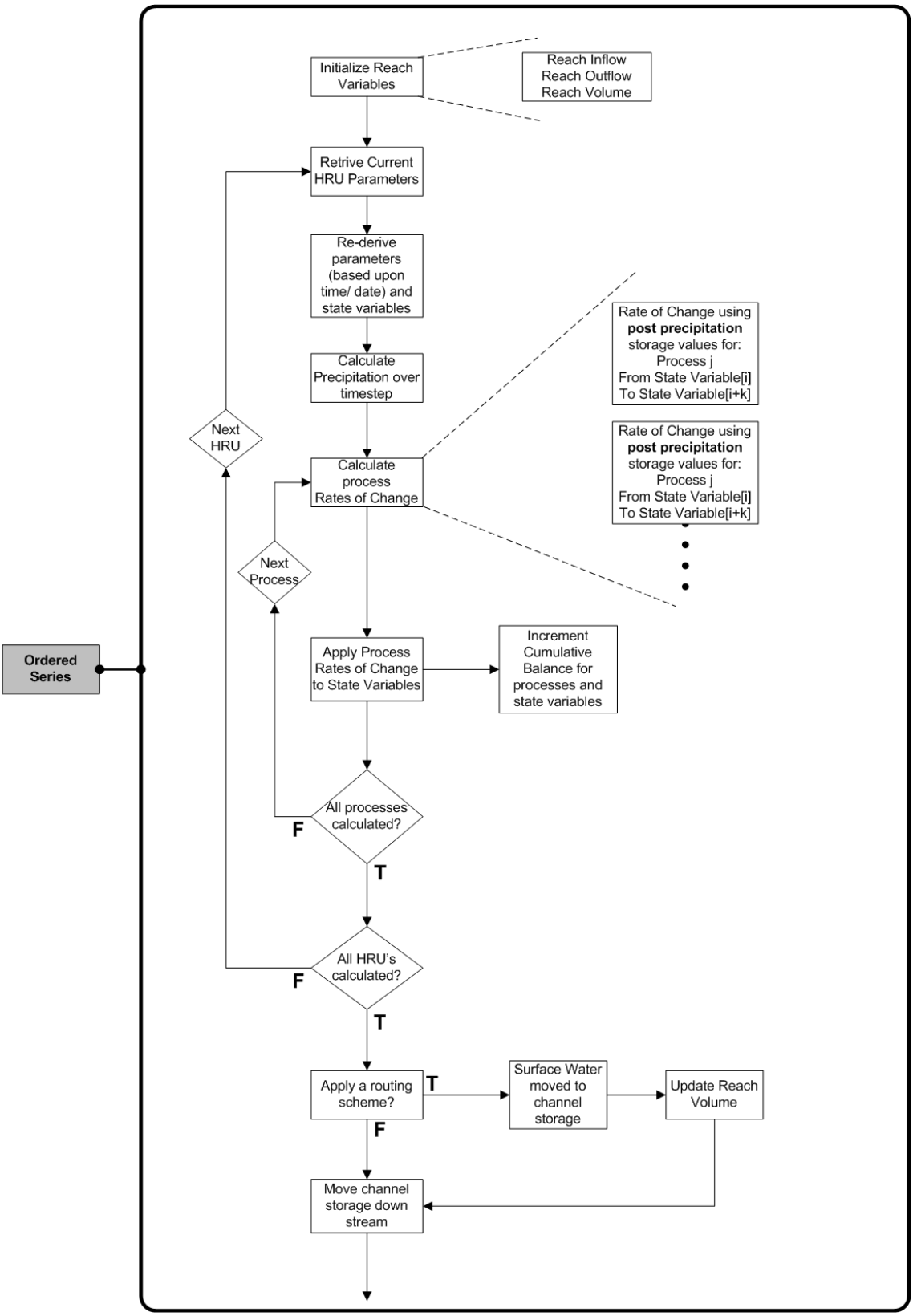

Figure 3.2: Flowchart of Ordered Series Approach in Raven

vantage of the Runge-Kutta methods is that they are slightly more computationally costly than the ordered series method. The advantages of the Runge-Kutta methods include the ability, with varying degrees of accuracy, to better approximate at larger timesteps the exact solution of the coupled ordinary differential equations that govern the system.

The Euler method, the lowest order Runge-Kutta method, calculates the rates of water redistribution between storage units based upon storage unit values at the start of the timestep and does not move any water or energy until all the current timestep's process rates have been calculated. This approach more closely approximates for theoretically 'simultaneous' operation of the physical processes. Unlike the ordered series approach, the order of processes does not impact the results.

The Heun and iterative Heun methods are essentially the same solver. The only variation between the two Heun methods is that one iterates until some convergence criteria is met, whereas the other iterates only once. The iterative Heun method convergence criteria can be specified by the user in order to increase the accuracy of the solution as needed.

3.2.2.2 Implementation Strategy

Using the notation from equation 3.1, the Euler method may be expressed mathematically as:

$$\{\phi\}^{t+\Delta t} = \{\phi\}^t + \Delta t \left([M]^t + \{Q\}^t\right) \tag{3.17}$$

Note that the matrix of process fluxes, $[M]$, is calculated solely from information at the start of the timestep. This equation, to solve for multiple ODEs, corresponds to equation 2.7, which is for a single ODE.

As can be seen in figure 3.3, the application of the Euler method is straightforward and similar to the implementation of the ordered series approach. Differences between the ordered series method and the Euler method include the use of pre-precipitation storage unit values and the simulation of concurrent physical process solutions when calculating the Euler approximation. This approach allows each process to use the same storage unit values (from the beginning of the timestep) for the solution approximation. Unfortunately, the Euler method is subject to the potential violation of thresholds which may results from the overdrawing of water from storage units.

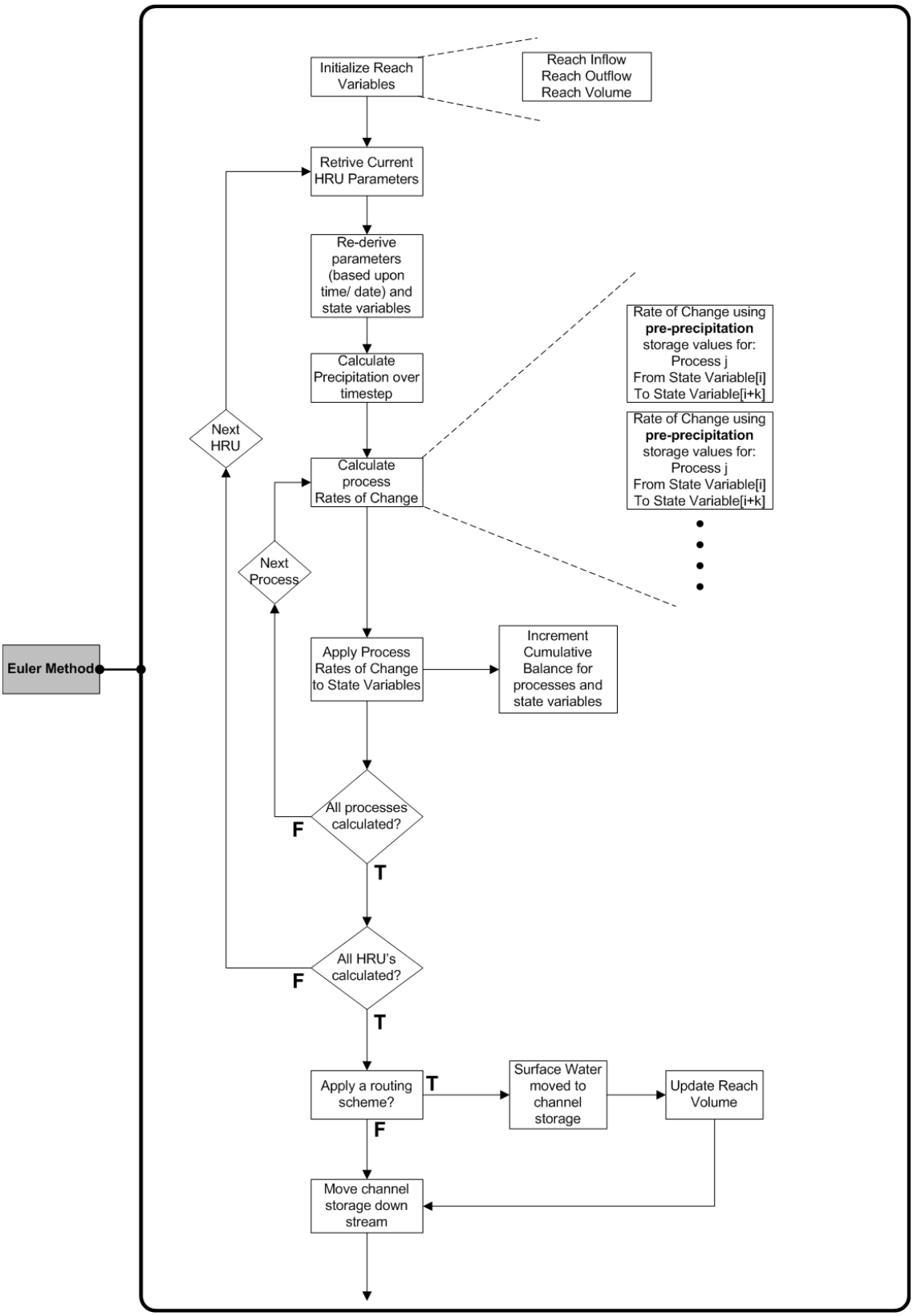

Figure 3.3: Flowchart of Euler's Method

The iterative implicit Heun method is a 2nd order Runge-Kutta method that calculates process rates of change by using information from both the beginning of the timestep and the end of the timestep. The iterative Heun method is solved by making an initial guess of the solution (the Euler solution) that is repeatedly refined. The user may either specify the level of precision required for the approximation provided by the iterative Heun method or specify a fixed number of iterations. The iterative Heun method can be expressed mathematically as:

$$
\begin{aligned}
\{\phi\}^{G} =& \{\phi\}^{t} + \Delta t \left(\left[M\left(\{\phi^{t}\}\right) \right]^{t} + \left\{ Q\left(\{\phi^{t}\}\right) \right\}^{t} \right) \\
\{\phi\}_{k=1}^{t+\Delta t} =& \{\phi\}^{t} + \\
& \tfrac{1}{2}\Delta t \left(\left[M(\{\phi^{t}\}) \right]^{t} + \left[M(\{\phi^{G}\}) \right]^{t} + \{Q(\{\phi^{t}\})\}^{t} + \{Q(\{\phi^{G}\})\}^{t} \right) \\
\{\phi\}_{k+1}^{t+\Delta t} =& \{\phi\}^{t} + \\
& \tfrac{1}{2}\Delta t \left(\left[M(\{\phi^{t}\}) \right]^{t} + \left[M(\{\phi_{k}^{t+\Delta t}\}) \right]^{t+\Delta t} + \{Q(\{\phi^{t}\})\}^{t} + \{Q(\{\phi_{k}^{t+\Delta t}\})\}^{t+\Delta t} \right) \\
\text{Until} \quad & \|\{\phi\}_{k+1}^{t+\Delta t} - \{\phi\}_{k}^{t+\Delta t}\| \leq \epsilon
\end{aligned}
\tag{3.18}
$$

where ϵ is a user specified convergence criteria. Equation 3.18 is the equivalent matrix form to the simple Heun method that was discussed in the section 2.2.2.

Figure 3.4 shows how the iterative implicit Heun method has been implemented within Raven. Similar to the Euler method, the iterative Heun method uses the storage unit mass/ energy values from the start of the timestep to calculate the initial 'rates of change' guess, denoted by superscript (G). The refinement of that guess is then calculated using the timestep's initial state variable values, the updated state variable values, and the previous iterations state variable values. These values are then used to calculate new redistribution rates for the physical process representations. When the new redistribution rates have been applied to the initial storage unit values, these storage unit values are compared to the previous iterations storage unit values to check for convergence. If the convergence criteria, as specified by the user has been met, the simulation can move to the next HRU or timestep. During these simulations, a maximum number of iterations can be set. For this thesis, 30 iterations was permissable. In the yearly simulations discussed in chapter 4, the average number of iterations required to reach convergence was three and the maximum number of iterations was only reached twice during the simulations.

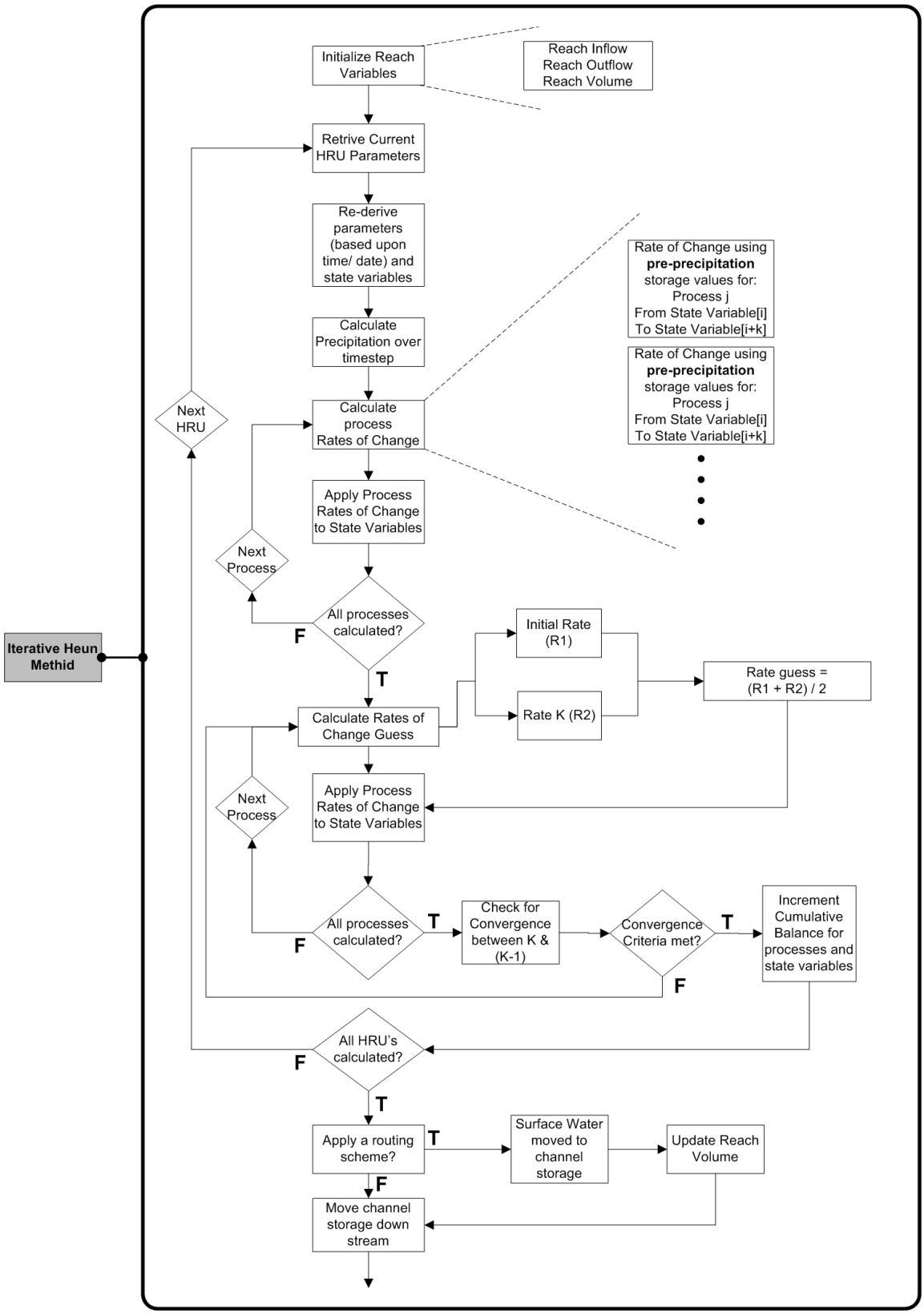

Figure 3.4: Flowchart of Iterative Heun Method

3.2.2.3 Computational Considerations

The Euler method is subject to global mass balance errors as timestep size increases because the Euler method does not explicitly recognize the influence of parallel processes for ODEs with continuous coefficients. For the same reasons, threshold constraints may not be met (e.g., storage quantities may become negative). As the timestep size decreases, both the ordered series and the Euler method increase in accuracy and will theoretically approach the exact solution to the mathematical problem statement.

For any given timestep, the iterative Heun method should be more accurate than the Euler method and the ordered series approach although it is the most computationally expensive due to the increased number of terms used from the Taylor series expansion [Hoffman, 1992]. However, it should also be more accurate at larger timesteps than the other methods. The solution will have higher accuracy if the order of the solver is increased or the use of finer timesteps is employed. Unfortunately, regardless of the solver method used, thresholds are still a problem.

3.3 Mass Balance Accounting

Mass balance accounting within distributed hydrological models is essential in order to verify that no mass or energy is lost from the system during calculations. Usually, in hydrological models, the global mass balance is the only item used to assess the numerical method robustness, even though it is easy to preserve the global mass balance. Here, global mass balance accounting calculations are performed at every timestep step to evaluate any potential mass balance errors which might indicate that programming, input or logic errors have occurred. This simple control method allows the modeller to verify that the system is maintaining an equilibrium of mass and energy. This method is calculated using the following formula:

$$\xi_{MB} = P_{cum} - T_{sw} - Q_{cum} + S_{init} \tag{3.19}$$

where ξ_{MB} is the mass balance error, P_{cum} is the cumulative precipitation over the modeled watershed$[L]$, T_{sw} is the sum of all mass/ energy in system $[L]$, Q_{cum} is the cumulative outflow from system $[L]$ and S_{init} is the initial system mass/ energy storage $[L]$. For this thesis, calculation of energy balance was neglected

because none of the physical process algorithms nor storage compartments used during testing were defined by their energy content.

Mass and energy can be lost several ways within a system and it is essential to constantly monitor to ensure that any losses are caught. Raven has several safeguards built into the model that prevent loss of mass and/or energy from the system. Global mass is always conserved in the system when modelled with Raven.

Numerical errors, such as operator splitting errors, are reflected in local mass balance errors, such as improper distribution of mass. These errors represent failures of the model solver to properly calculate and redistribute the mass and energy of the system.

Chapter 4

Results and Discussion

4.1 Site Description & Model Configuration

The site that was used as a basis for testing is the Nith River Watershed located in southwestern Ontario. It is a sub-watershed of the Grand River Watershed, as shown in figure 4.1. A detailed map of the Nith River watershed is shown in figure 4.2. Only the portion of the Nith River that is located upstream of the New Hamburg stream gauge (02GA018), located at 43°22′29" North and 80°42′41" West, was included in the models. Land use, vegetation cover and soil strata used to construct the Nith River Watershed model are summarized in table 4.1, table 4.2 and table 4.3 [Modified from Presant and Wicklund [1971]].

	Agri.	**Forest**	**Wetland**	**Urban**	**Bare**	**Water**	**Total**
Area(km^2)	466	50.0	12.2	8.8	0.4	14.3	552
Percent	85	9	2	1	<1	2	100

Table 4.1: Land use for the Nith River Watershed [Lang, 2008]

Forest Type	**% Area Covered**
Broadleaf	20
Conifer	80

Table 4.2: Nith River Watershed Forest Types

The information summarized in tables 4.1 - 4.3 and figures 4.1 - 4.2 has been used to build the model and input files for a one-year continuous simulation of 2004

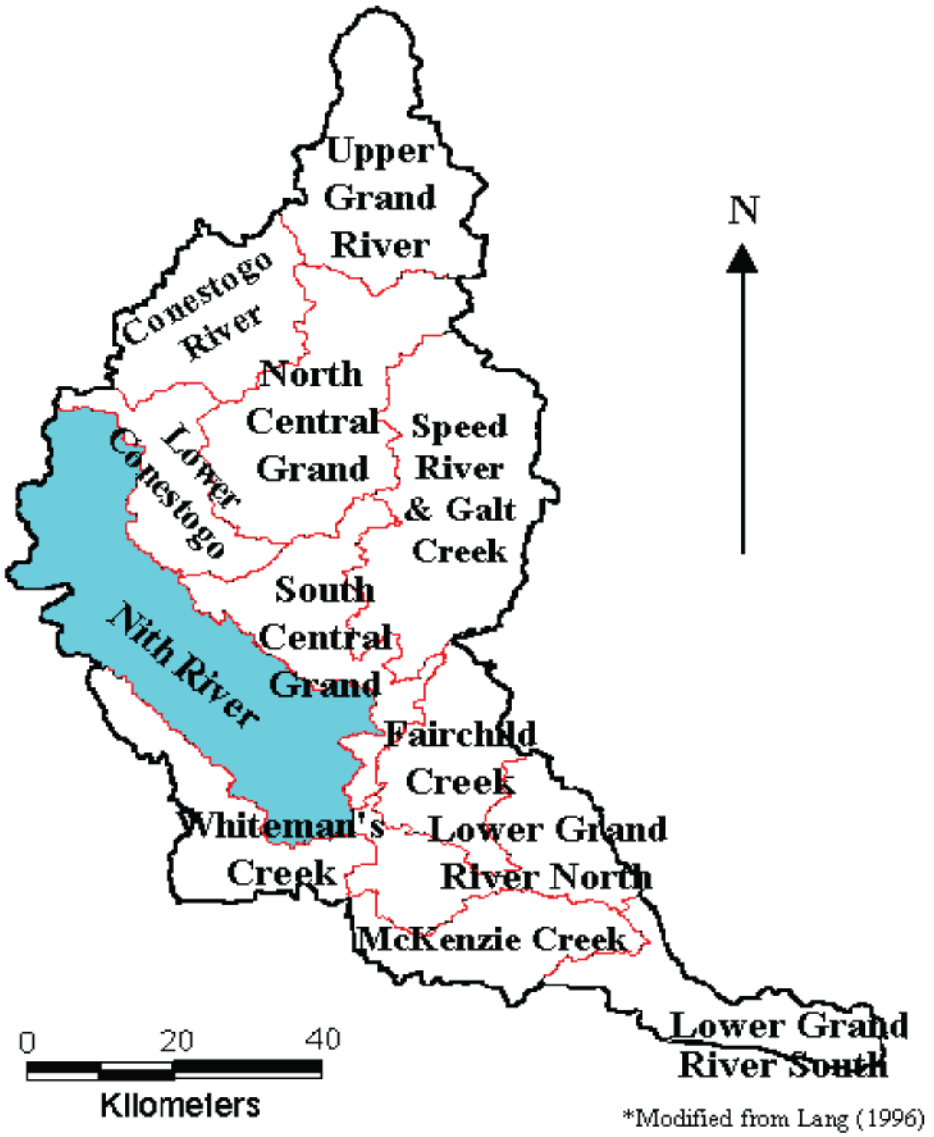

Figure 4.1: Grand River Watershed and Subwatersheds [Lang, 2008]

Soil Layer	Depth [cm]	Sand [%]	Silt [%]	Clay [%]	Organic [%]
Layer 1	0-10.16	23	41	36	7.0
Layer 2	10.16-20.32	27	41	32	1.8
	20.32-30.48	23	41	36	0.8
	30.48-45.72	20	42	38	0.9
Layer 3	45.72-60.96	14	40	46	0.6
	60.96+	19	45	36	0.4

Table 4.3: Nith River Watershed Soil Profiles

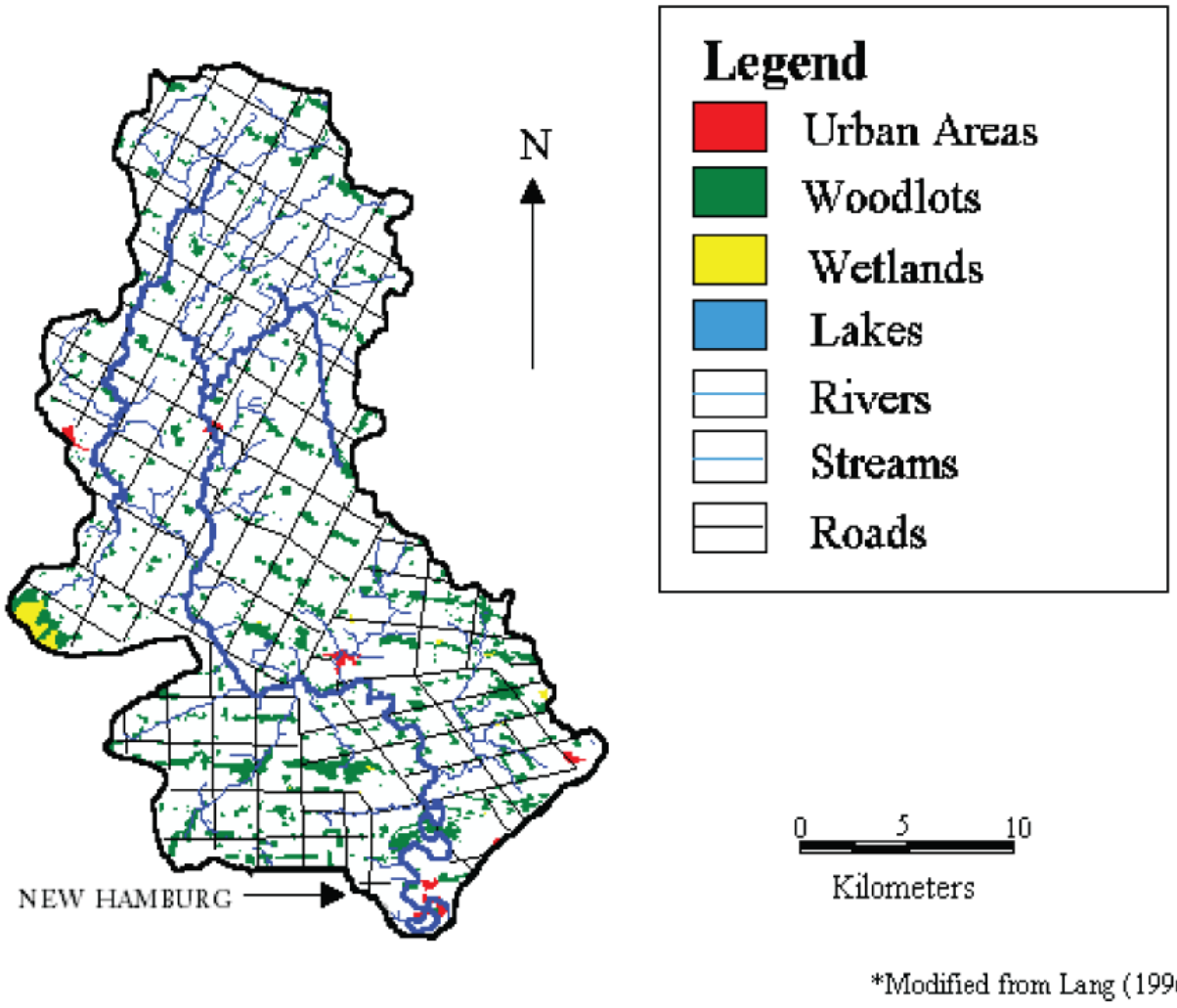

Figure 4.2: Landuse of the Nith River Watershed

with Raven. The four largest land classes (i.e., agricultural, forest, wetland and urban) in the Nith River Watershed have been used to define the 4 HRUs for these tests. Many aspects of the basin have been inferred based on the information of Lang [2008] and Presant and Wickland [1971] for use in model calibration. Raven calculates several derived watershed parameters which helps minimize the amount of data required to assemble the input files. The timeseries data for precipitation and temperature was provided by the University of Waterloo Weather Station located in Waterloo, Ontario at 43°28′25.6" North and 80°33′27.5" West [Seglenieks, 2008]. All data used for these simulations were at daily intervals.

It is important to note that, for the purpose of this thesis, the physical accuracy of the model is not considered as important as the models mathematical precision.

4.2 Experimental Design Overview

Multiple simulations were performed to analyze the impacts and significance of using different numerical algorithms to simulate water fluxes within the Nith River watershed. Three numerical methods were tested: the ordered series method, the explicit Euler method and the iterative implicit Heuns method. All simulations were run at daily and 5 minute time intervals. Additional intermediate time intervals were used to assess solution convergence and error bounds.

Tests were initially used to establish mathematical 'truth'. 'Truth' is represented by the 5 minute timestep simulations because most of the numerical methods tested converge to a single answer at this level of discretization. Further tests were then conducted to compare diagnostic results from different algorithms at a range of temporal resolutions to mathematical 'truth'. These tests focused on multiple operation orders with the ordered series method and the use of the Euler and iterative Heuns methods.

Three different forms of model output were used to evaluate model performace: storage, cumulative redistribution and hydrographs. Storage results show the amount of mass or energy present in a state variable or storage unit at the end of every timestep. Cumulative redistribution results show the cumulative mass or energy that has been moved by a specific process at the end of every timestep. Hydrographs show the channel, river or stream flow exiting the watershed at the end of each timestep. Comparisons of storage, cumulative redistribution and hy-

drograph results for multiple numerical methods and multiple timesteps are shown in the following sections.

Two test cases were used to simulate the Nith River Watershed. Test case 1 (TC1) is a calibrated simulation and test case 2 (TC2) is a simplified simulation. The state variables, processes and parameters depicted in figure 4.3 represent TC1. Test case 2, depicted in figure 4.4, is nearly identical to test case 1, but differs in its parameterization. The purpose of test case 2 is to emphasize the potential errors that may be generated in hydrological models due to operation order. Using these two test cases, four primary steps were taken to meet the primary objectives of the thesis. Initially the model was calibrated to Nith River gauge data. Following calibration of this model configuration (TC1), the impacts of multiple ordered series approaches at several timesteps and the performance of the Runge-Kutta methods at several timesteps were investigated. Comparisons between the ordered series approaches and the Runge-Kutta methods were then conducted to evaluate their relative performance. Since the calibrated model demonstrated minimal operator splitting errors for this watershed, the TC1 model was simplified to create TC2 by reducing the number of storage compartments and hydrological processes as well as manipulating some of the process input variables, which still represent the Nith River hydrology but provides a better, more error-sensitive model for comparing algorithms. Following these tests, an error analysis was performed to quantify the mathematical accuracy of each numerical method. These steps are discussed in detail in the following sections.

4.3 Model Calibration

Raven was auto-calibrated with the Ostrich calibration software [Matott, 2005] to the New Hamburg stream gauge data. The goal was to modify watershed, HRU and process parameters so that Raven's output would reasonably match the Nith River gauge data. Calibration serves to show that Raven is capable of replicating a real world location with confidence. A perfect replication of the Nith River was not the goal, only a reasonable representation was desirable. The iterative Heuns method at a daily timestep was used for calibration. In order to calibrate Raven to match the Nith River gauge data, it was necessary to introduce a ponded water storage compartment (i.e., depression storage). The ponded water storage compartment receives a portion of all precipitation, while the depression is not full, and is subject

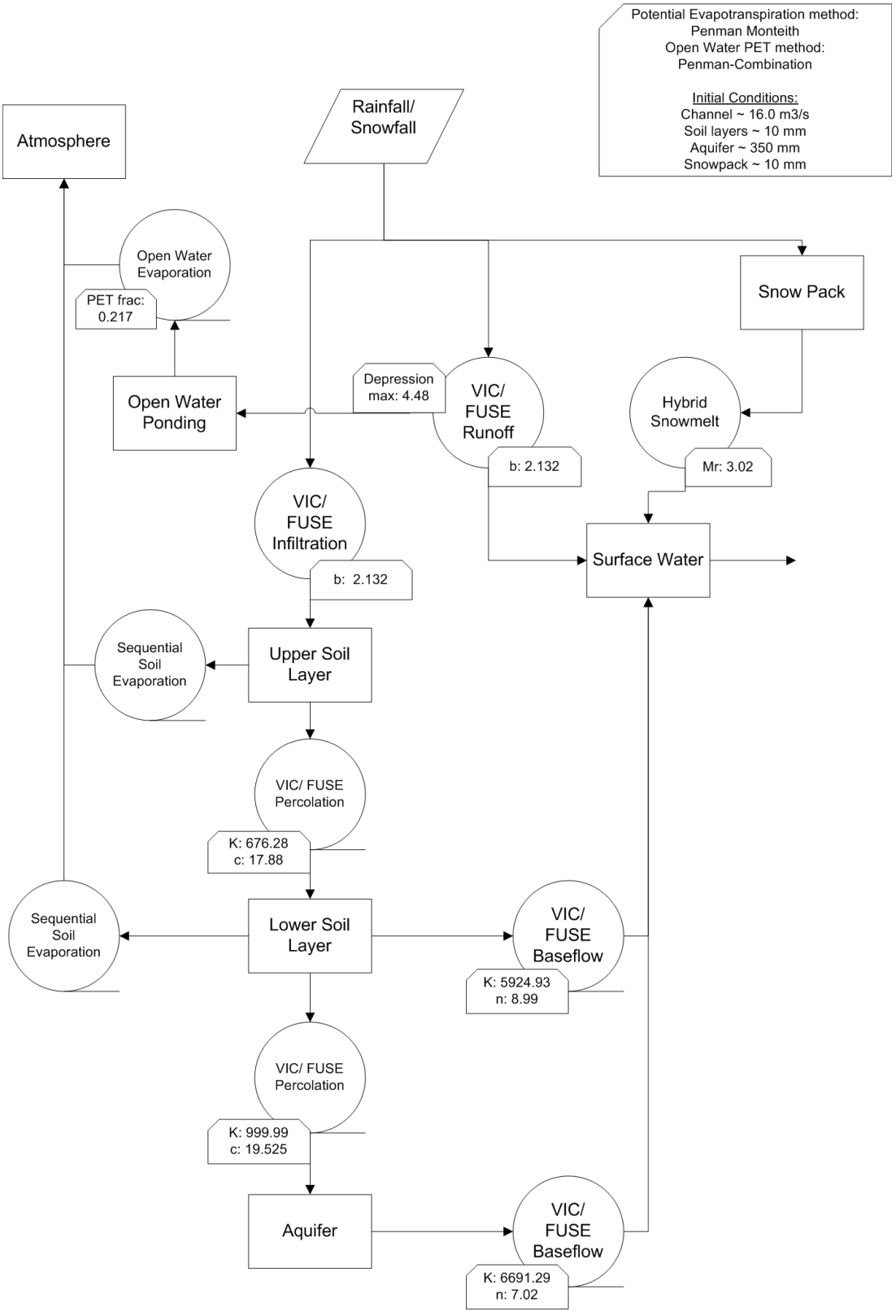

Figure 4.3: Flowchart of Raven TC1 showing state variables, processes and calibration parameters

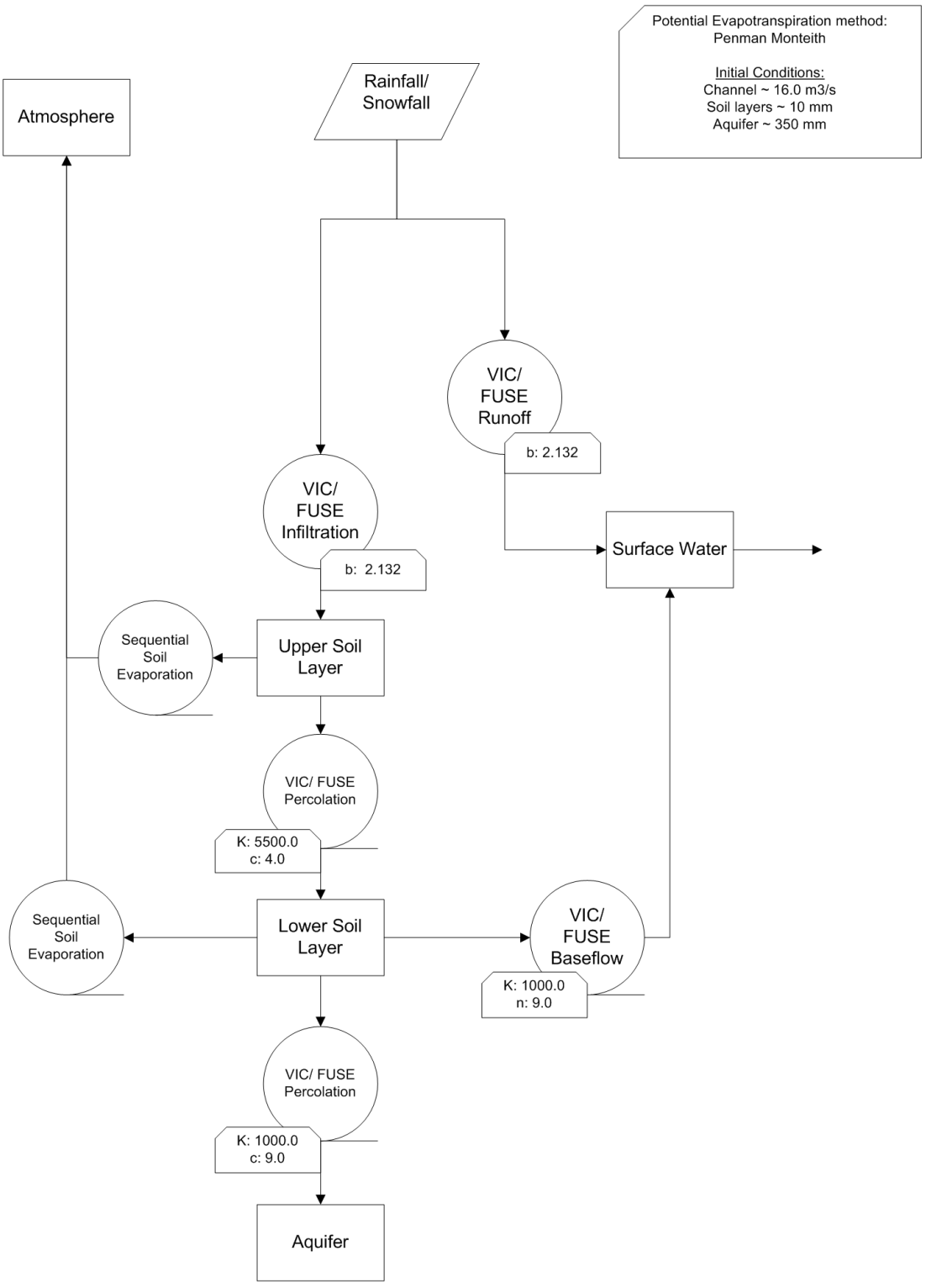

Figure 4.4: Flowchart of Raven TC2 showing state variables, processes and calibration parameters during operator order investigation

only to evaporation. The water moved to ponded water storage does not enter the surface water store and therefore does not get routed through the river/ channel storage unit. Post-calibration hydrographs showing daily streamflow (figure 4.5) and 5 minute timestep streamflow (figure 4.6) for the ordered series approach, Euler method and iterative Heuns method are discussed in this section.

The simulated results show that using TC1, Raven is able to produce streamflow that is comparable to the measured flow of the Nith River at the daily timestep with a Nash-Sutcliffe of -0.10. With this configuration, Raven appears to have difficulties replicating the streamflow during storm events and the resulting hydrographs depicts more minor streamflow fluctuations than the gauge data during the course of the simulation. Similarly, the simulated streamflow produced by Raven is not able to replicate the initial melt at the end of winter. These errors are likely due to a lack of appropriate physical process repesentations, but could also be due to inappropriate discretization of the simulated watershed. In spite of this, Raven has still managed to produce reasonable hydrographs (figure 4.5) using all numerical methods.

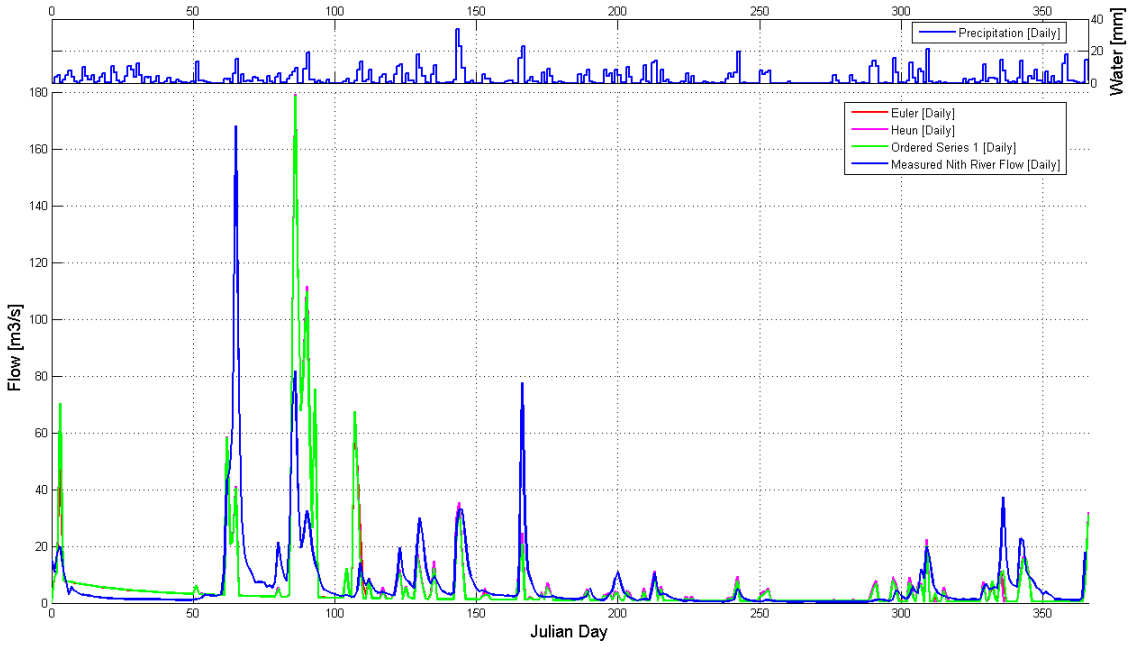

Figure 4.5: Hydrograph from TC1 showing the ordered series, Euler and iterative Heuns method compared to the Nith River Stream Gauge at a daily timestep

Using the same parameters as obtained from calibration using the iterative

Heuns method, the ordered series, Euler and iterative Heuns methods generate hydrographs that are essentially identical with some minor variations between them at peak streamflow events. Minor variations can be seen in figure 4.6, where the simulation range is constrained to 100 days. However, it is important to note that the 3 different numerical methods produce essentially identical hydrographs and that those hydrographs are comparable to the measured streamflow from the Nith River. A five minute timestep was used for the simulations here. With such a small timestep, these numerical methods are not expected to produce different results.

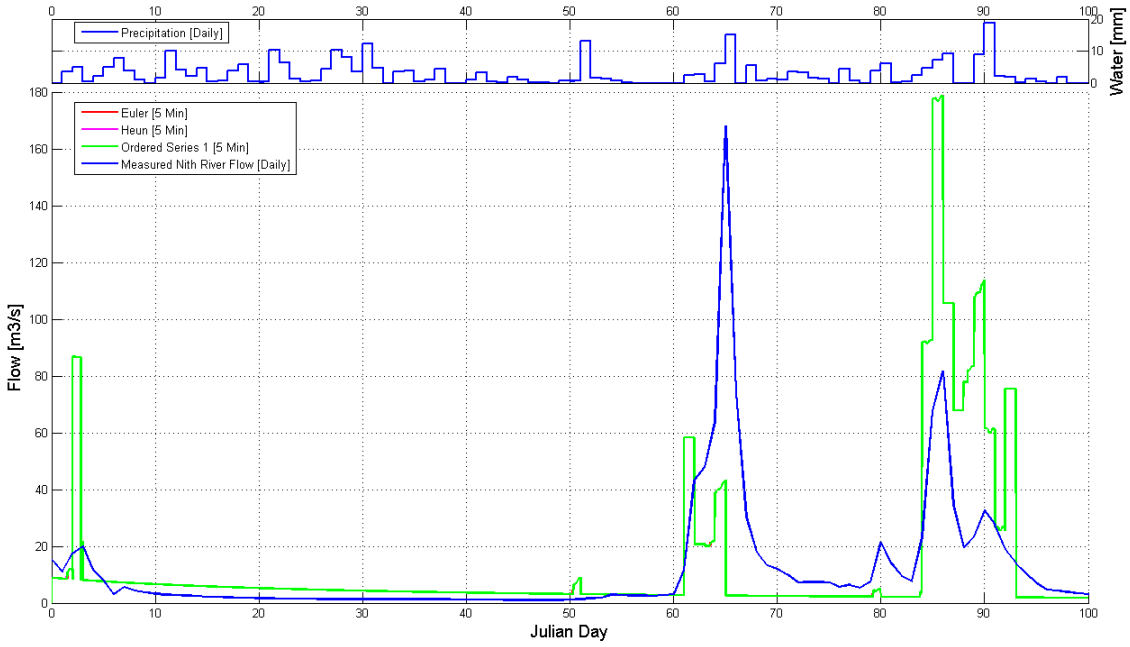

Figure 4.6: Hydrograph from TC1 showing the ordered series, Euler and iterative Heuns method compared to the Nith River Stream Gauge using a 5 minute timestep

4.4 Ordered Series Approach

It is hypothesized here that operator order in the conventional ordered series approach will impact model results, and that this impact may worsen with increasing timestep size. To test this hypothesis, both TC1 and TC2 were used. Test case 1 provides a reasonable replication of the Nith River Watershed and test case 2 was used to accentuate the potential competition that exists between hydrological processes. These simulations were compared to results from the finely discretized sim-

ulations using all numerical methods. When the simulations are finely discretized, results from all numerical methods converge to the same solution.

4.4.1 The Influence of Operator Order

Most hydrological models use an ordered series approach, without regard for the potential impact of operator order. Here, six orders of operation (table 4.4) were used to simulate the Nith River Watershed with TC1. Errors due to operator order have been assessed by examining simulated output hydrographs (figure 4.7) and simulated storage of the upper soil layer (figure 4.8). Six orders of operation (table 4.5) were also used for TC2.

	Hydrologic Process
A	Precipitation
B	Soil Evaporation
C	Percolation (Upper Soil to Lower Soil)
D	Percolation (Lower Soil to Groundwater)
E	Baseflow (Lower Soil to Surface Water)
F	Baseflow (Groundwater to Surface Water)
G	Snowmelt
H	Open Water Evaporation
Numerical Method	**Process Order**
Ordered Series 1	A B C D E F G H
Ordered Series 2	A H D G B C F E
Ordered Series 3	A E H D B F C G
Ordered Series 4	A G E H C B F D
Ordered Series 5	A H G B F E D C
Ordered Series 6	C D B G E H F A
Euler Method	Not applicable to this method
Heun Method	Not applicable to this method

Table 4.4: Hydrological process labels and order of physical process representations for TC1 ordered series testing. Process order does not affect results of Euler and Heun method simulations.

It can be seen in figure 4.7 that all TC1 operator splitting approaches generate nearly identical hydrographs at the daily timestep. It has already been stated that one of the reasons that the ordered series approach is favored in hydrological models is due to its ability to perfectly conserve mass. Since the hydrograph represents the total net outflow from a system dominated by runoff, it is not surprising that all six operation orders produce similar hydrographs. This is due to the fact that the

	Hydrologic Process
A	Precipitation
B	Soil Evaporation
C	Percolation (Upper Soil to Lower Soil)
D	Percolation (Lower Soil to Groundwater)
E	Baseflow (Lower Soil to Surface Water)
Numerical Method	**Process Order**
Ordered Series 1	A B C D E
Ordered Series 2	A C E B D
Ordered Series 3	A D B E C
Ordered Series 4	A E D C B
Ordered Series 5	C E B D A
Ordered Series 6	E D C B A
Euler Method	Not applicable to this method
Heun Method	Not applicable to this method

Table 4.5: Hydrological process labels and order of physical process representations for TC2 ordered series testing. Process order does not affect results of Euler and Heun method simulations.

model simulation redistributes water between storage compartments with surface water/ channel flow being the end of the water pathway.

Figure 4.8 depicts the water storage in the upper soil layer from a one year simulation of TC1 using a daily timestep. Results only varied slightly in the upper soil layer during the simulation due to operation order. As can seen in the graph, only ordered series 6 is visually distinguishable from the results of the other five ordered approaches. This is likely due to the precipitation input being added to the simulation at the end of a timestep instead of at the beginning as is done in the other five series. Due to the precipitation input at the end of the timestep, ordered series 6 has less water available for redistribution by the hydrological processes during a timestep, which results in ordered series 6 lagging a day behind the other orders when simulated.

The simulated output hydrograph (figure 4.9) of TC2 shows that each ordered series approach produces different results. The output from all six ordered series follows the same general trends but each operation order produces a unique time series. The critical difference between TC1 and TC2 that influences this behaviour is the speed and demand of the hydrological processes. In TC1, there is limited competition between processes which results in the more demanding process redistributing the majority of the available water. In TC2, the parameters that define

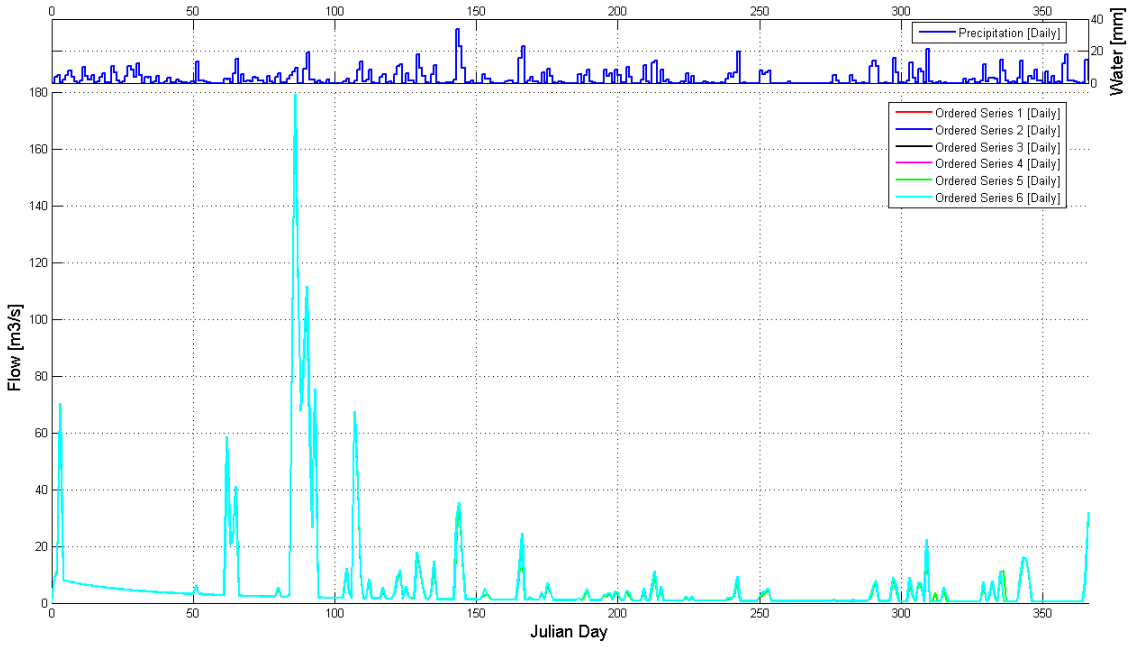

Figure 4.7: Hydrographs from TC1 for multiple ordered series approaches during a one year simulation

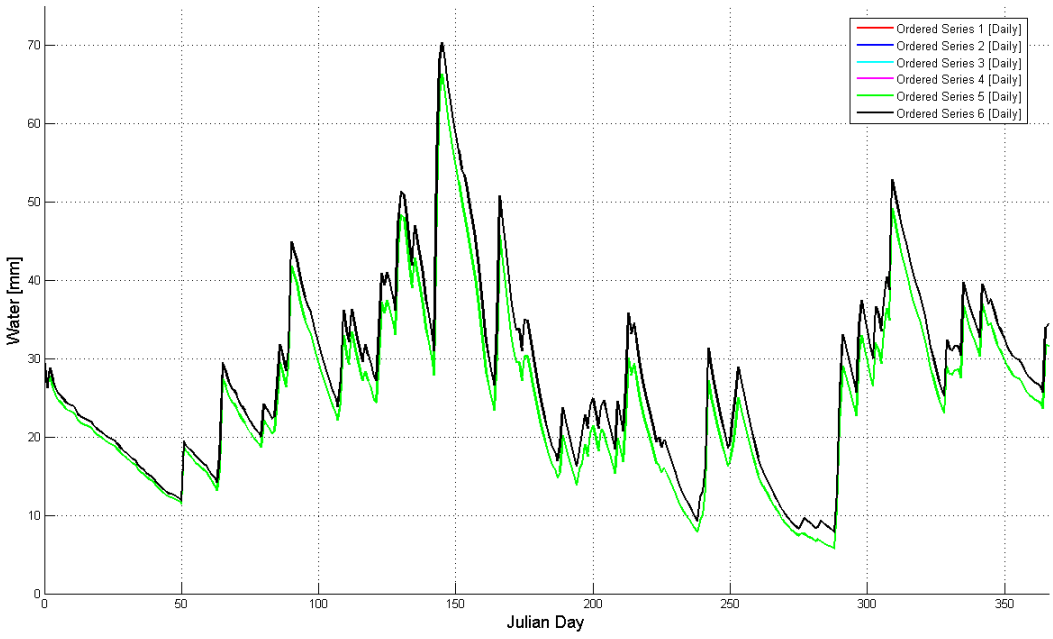

Figure 4.8: Upper soil layer storage from TC1 for multiple ordered series approaches at a daily timestep

process rates have been altered to make the redistribution rates of all processes, interacting with the same storage unit, nearly equivalent and of high demand. This creates a highly competitive environment which shows how operator order may affect hydrological model results.

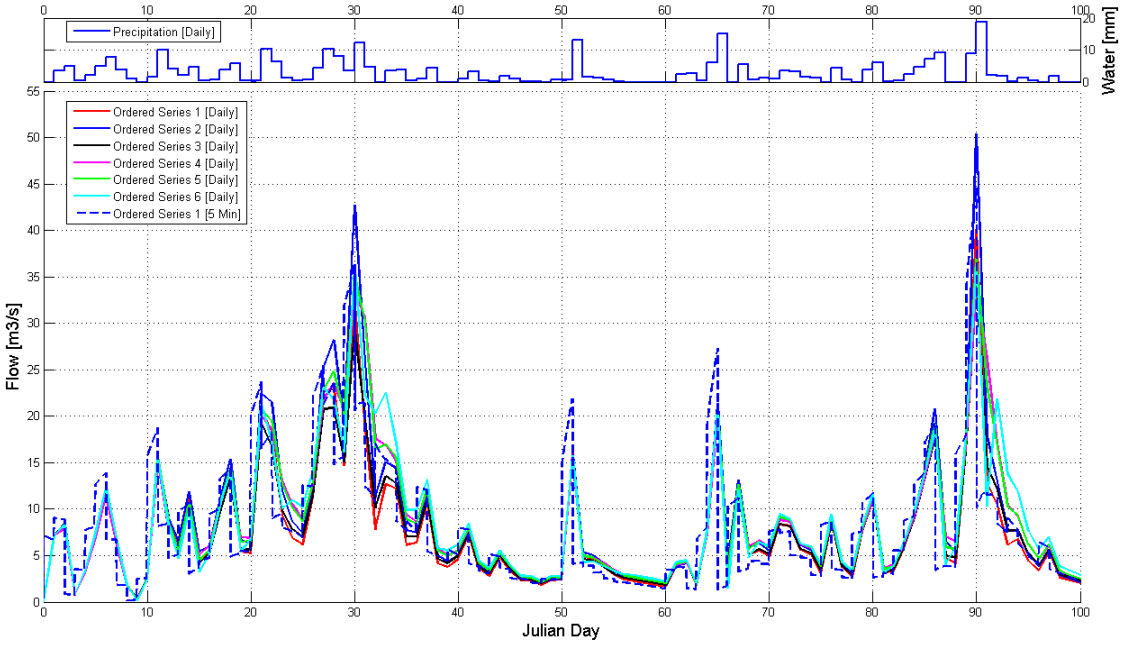

Figure 4.9: Hydrographs from TC2 for multiple ordered series approaches at a daily timestep

Simulated storage in the upper soil layer from TC2 shows higher variability in storage. Again all six operation orders produce unique results. The water content of the soil layer fluctuates continuously and all six ordered series approaches follow the same trend. Similar results are found for the lower soil layer.

Both the upper and lower soil layers share interesting traits: the two ordered series cases where precipitation was added at the end of the timestep (ordered series 5 and 6) are producing results that are not the same values. Ordered series 6 tends to generate higher soil water content then the other ordered series approaches and ordered series 5 tends to generate the lowest soil water content during the first half of the simulation. This trend reverses in the second half of the simulation. This variation in the results from the ordered series approaches is due to the effects and sensitivity of the hydrological processes used in the simulation. These differences are minor for any actual modelling application, however that may be different

depending on the simulated watersheds location and parameters. Due to the fact that the effect of operation order on results is not easy to predict, ordered series approaches have the potential to generate unexpected results.

4.4.2 Impact of Timestep on Operation Order Errors

The previous tests examined only the impact of operator splitting errors. Additional tests were performed to examine how timestep size influences these errors. Six orders of operation were used during Raven's simulation of the Nith River Watershed at a daily and 5 minute timesteps to identify how closely the six ordered series are able to replicate 'mathematical truth', represented by model results from a 5 minute timestep run of ordered series 1.

TC1 was used to produce simulated streamflow (figure 4.10), cumulative losses to the atmosphere (figure 4.11) and upper soil layer storage (figure 4.12) graphs that are used to evaluate the accuracy of the ordered series approaches as timesteps increase from 'truth' (5 minute) to daily. For comparison in this section, only the 5 minute and daily timesteps are evaluated. Additional timesteps are discussed in section 4.6.3.

Figure 4.10 shows a comparison of the six ordered series approaches, from TC1, at a daily timestep to results from ordered series 1 using 5 minute timesteps. The hydrograph timescale has been shortened to highlight the differences between the simulation outputs. As before, the six ordered series approaches at a daily timestep all produce similar results (as was shown in figure 4.7). As the timestep is decreased, the results from the ordered series tests converge.

The 5 minute timestep hydrograph in figure 4.10, representing 'truth', shows nearly identical results compared to the ordered series approaches that were calculated at the daily timestep. When the timestep shrinks from daily to the 5 minute, the output resolution improves. During baseflow moments (the 'normal' flowrate of the river when no precipitation events are occurring) of the hydrographs, timestep plays less of a role. This is seen on the relatively 'flat' sections of the hydrographs where the multiple timestep results overlap.

Cumulative losses to the atmosphere include evaporation from both soil layers and open water. As figure 4.11 depicts, ordered series 1 - 5 of TC1, produce identical results. Ordered series 6 shows slightly lower values due to the time of precipitation input. The results of the ordered series approach simulated at the 5 minute timestep

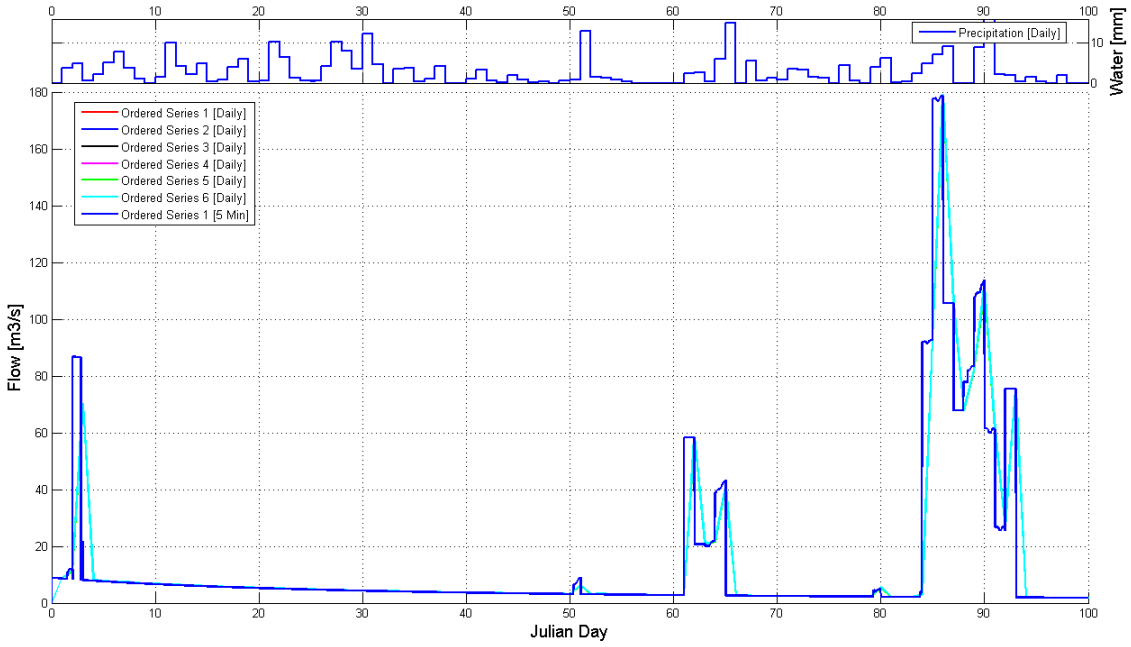

Figure 4.10: Hydrograph from TC1 for multiple ordered series approaches using daily and 5 minute timesteps

show higher evaporation than the daily timestep results. This difference in results is likely due to the fact that soil evaporation plays a large role in this model and is able to act on the soil layers at a faster rate than other processes in the system when at a smaller timestep. It is also likely that the depression storage, which is only acted upon by evaporation, receives a larger amount of water during the simulation with smaller timesteps due to the fact that open water evaporation acts quickly on the available ponded water. This then allows more water to be redistributed into the ponded storage compartment during precipitation events.

Figure 4.12 shows how variable the water content in the upper soil layer is over time. It is clear from this graph that the ordered series approaches (using daily timesteps) are not able to replicate the finely resolved model results.

During simulation of the lower soil layer with TC1, the ordered series approaches fluctuate and do not coincide with the finely resolved solution. Ordered series 1 - 5 generate results that are identical and ordered series 6 is nearly identical. The lower soil layer is emptying slowly regardless of incoming water. This is likely caused by the baseflow and percolation processes. Truth is simulated to be lower than the daily simulation. This is a good indication that ordered series approaches do not

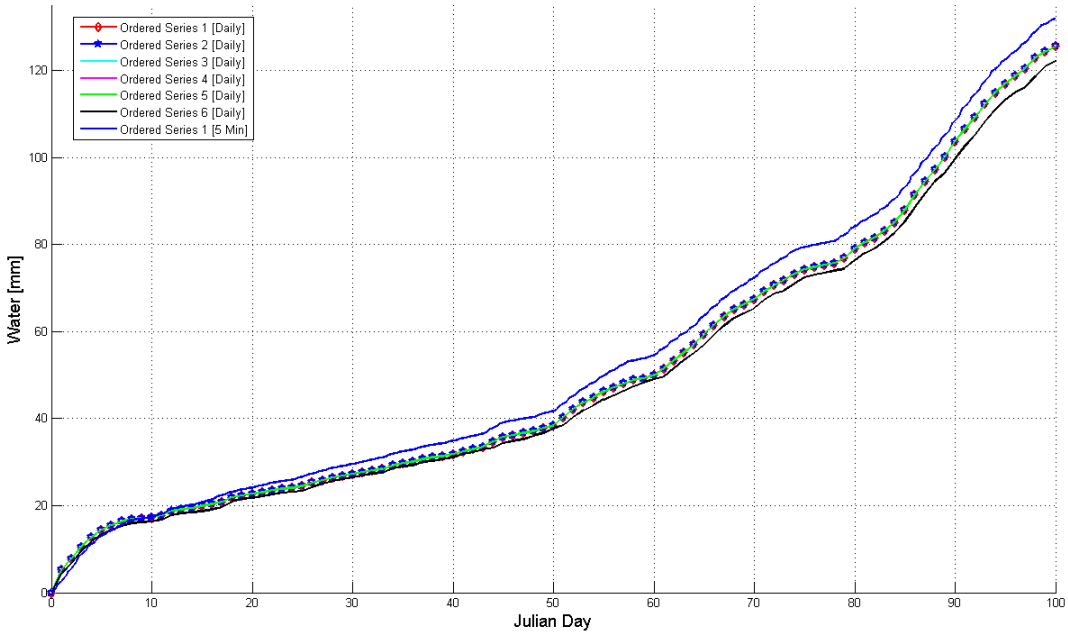

Figure 4.11: Cumulative losses to the atmosphere from TC1 for multiple ordered series approaches using multiple timesteps

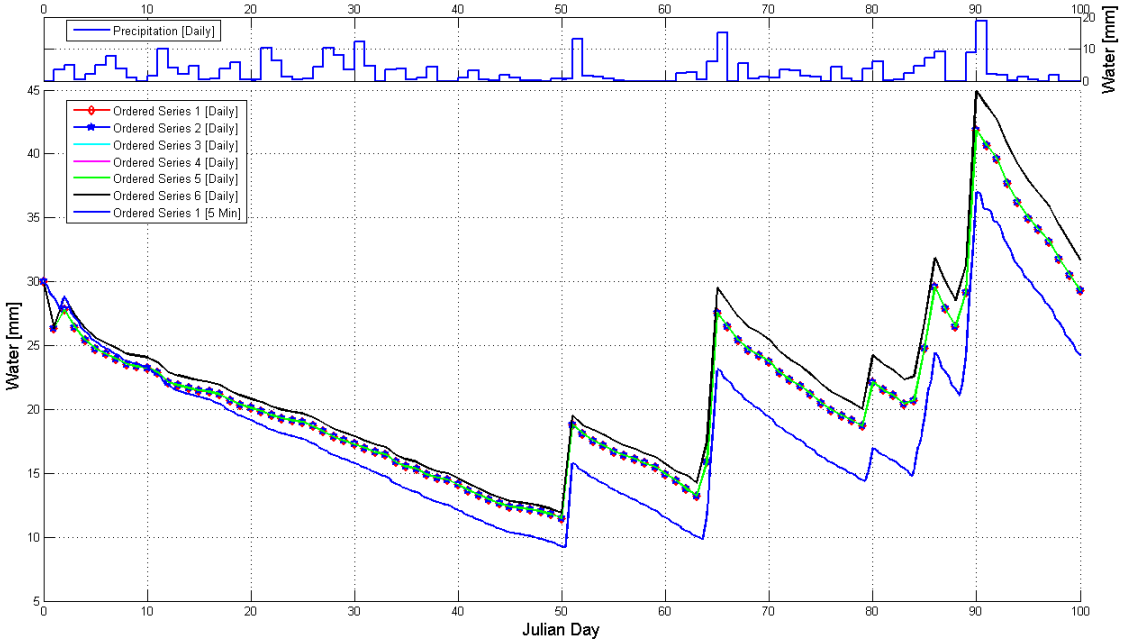

Figure 4.12: Upper soil layer storage from TC1 for multiple ordered series approaches using multiple timesteps

perform well at a daily timestep and that the use of some processes (e.g., baseflow, percolation) when used at the end of the operation order have the potential to continuously empty storage units.

The upper soil layer storage from TC2 shows that regardless of the operator order used, no daily representation is capable of replicating truth. All six operator orders produce different results, some of which are significantly different. Ordered series 5 and 6 are the closest to replicating truth through the majority of the simulation, which may be due to the precipitation being added to the system at the end of the timestep. Ordered series 1 - 4 generate similar, but not identical, results. Similar results are found in the lower soil layer (figure 4.13) using TC2.

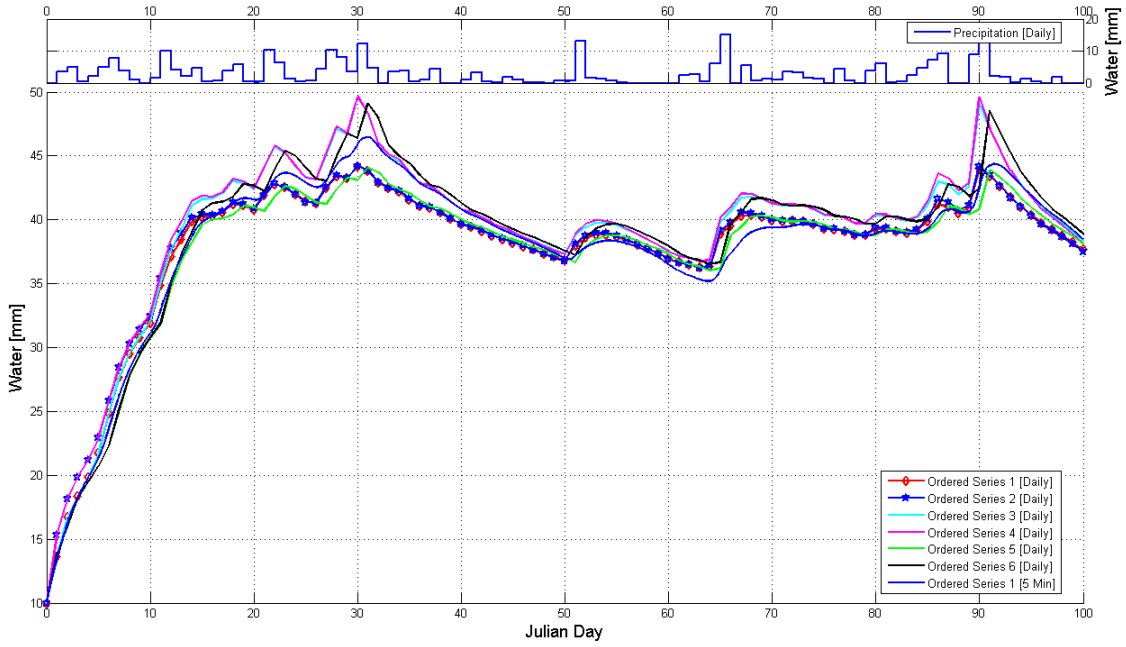

Figure 4.13: Lower soil layer storage from TC2 for multiple operator orders using two timestep sizes

4.4.3 Ordered Series Summary

The ordered series approach is used by most models to perform their redistribution rate calculations. The use of this method may incorrectly apply model processes which can theoretically result in poor model results and cause misinterpretations of the contributions of the storage units and redistribution fluxes in a simulated

watershed. Due to the variability of the results with series order, it would seem prudent to limit or avoid the use of ordered series approaches to solve models. This restriction is even more warranted when using a low time resolution which will exacerbate potential errors. Although the errors found in these tests were small, the potential for significant operator splitting errors in simulations raises concerns about the validity of results that can obtained through the use of hydrological models that use an ordered series approach as well as any decisions that may been made based upon those results.

4.5 Runge-Kutta Methods

It is thought that the use of Runge-Kutta methods will negate operator order impacts in hydrological models and reduce numerical errors at larger timesteps. To test this hypothesis, simulations in this section are intended to show that the use of numerical methods other than the ordered series approaches in distributed hydrological models can generate results that may be more accurate and unaffected by operation order. Assessment of the results from different Runge-Kutta methods (Euler, iterative Heuns) and the ordered series approach was conducted using daily and 5 minute timesteps.

4.5.1 Convergence Behaviour

Neither the Euler method or iterative Heuns method at a daily timestep is able to replicate the results produced at a 5 minute timestep, but it is apparent that the daily Euler and iterative Heuns method provide a close approximation of the finely resolved solution. The iterative Heuns method is more computational expensive (in theory) when compared to the Euler method regardless of timestep. The number of iterations used by the iterative Heuns method also affects the computational cost of the method. During these tests, the iterative Heun averaged three to four iterations per HRU per timestep before solution convergence occurred. A full discussion of the results from the Runge-Kutta methods including graphs is in section 4.6.

4.6 Numerical Method Comparison

The tests in this section were designed to test whether simulations at a daily timestep using Runge-Kutta methods, specifically the iterative Heuns method, will out perform the Euler method and the traditional ordered series approach for accuracy. Comparison of the results from both the ordered series and Runge-Kutta methods are used to highlight how performance varies based on numerical method choice. The comparisons were performed using both a daily and 5 minute timesteps.

4.6.1 Daily Timestep

The ordered series approaches, the Euler method and the iterative Heuns method were used at a daily timestep to see how the various numerical methods perform in comparison to each other. The hydrographs (figure 4.14) show that ordered series 1, the Euler, and the iterative Heuns method at the daily timestep produce almost identical results for watershed outflow. Ordered series 1 was chosen to represent the ordered series approaches because the simulations for ordered series 1, the Euler and iterative Heuns method were all conducted with the same input file that defines the order in which the global process solver applies the hydrological processes. The hydrographs confirm that all three methods are sending roughly equivalent amounts of water into the channel during the course of the simulation. As was shown in previous results for this model setup, there is little variation in hydrographs due to numerical method selection.

4.6.2 Multiple Timesteps

The tests in this section were completed using the ordered series approaches and the Runge-Kutta methods. The ordered series approaches, the Euler method and iterative Heuns method were applied to Raven's simulation of the Nith River Watershed at daily and 5 minute timesteps to compare the three methods to each other in order to assess their viability, at the daily timestep, to provide a stable and mathematically accurate solution.

Figure 4.15 depicts the results generated for cumulative open water evaporation from TC1. The 5 minute timestep simulations all produce the same results which therefore overlap on the graph. This is likely due to the fact that the calibration for

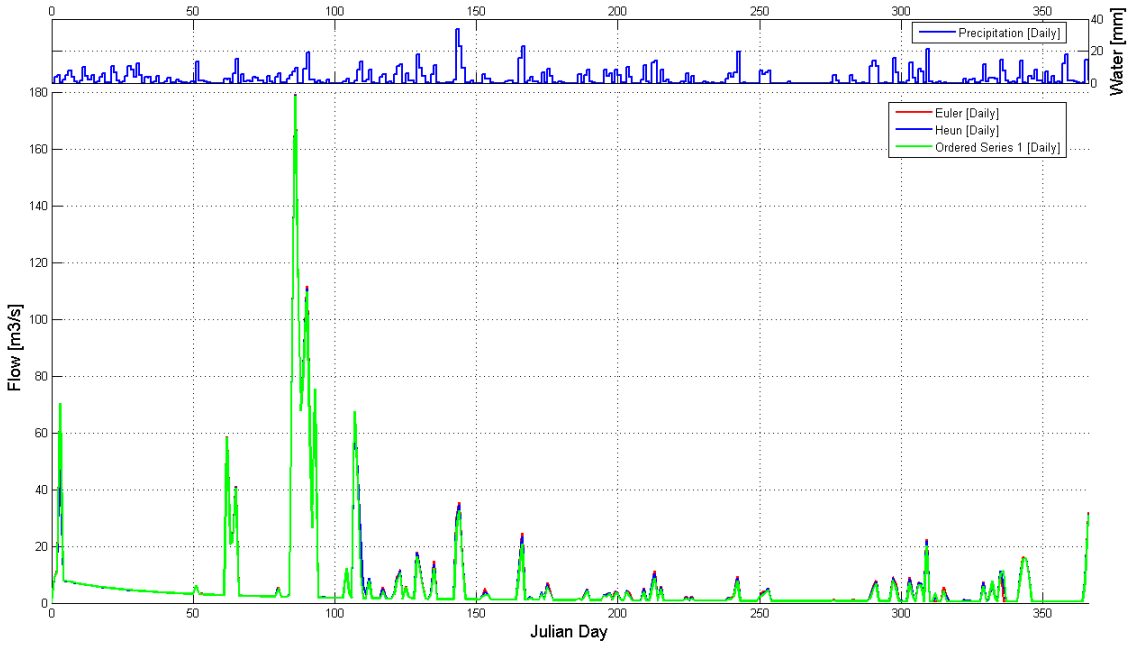

Figure 4.14: Hydrograph from TC1 of simulations using ordered series 1, Eulers method and iterative Heuns method at a daily timestep

this particular watershed is composed of hydrological processes that do not compete equally. The faster processes (e.g., evaporation) take the majority of the available water regardless of where the process occurs in the operation order. As the graph shows, at the end of a 100 day simulation, there is already a 20% difference in the cumulative open water evaporation amounts. This may increase further if the simulation were to run for a longer duration. This is a good example of a situation where the ordered series approach may negatively impact decisions and predictions.

Figure 4.16 provides a comparison between the six ordered series methods and the Runge-Kutta methods using TC2. As stated previously, at the 5 minute timestep, solutions from all methods converge to the same solution. This figure shows that the six ordered series approaches have large variability in their results. Ordered series 1 - 4 produce nearly identical results, where series 1 and 3 overlap and series 2 and 4 overlap on the graph. Series 1 and 3 show slightly higher values than series 2 and 4. Ordered series 5 and 6 overlap and are partially hidden behind the results given by the Euler and iterative Heuns methods. As is visible on the graph, the daily timestep Euler and Heuns results are nearly identical to the 5 minute timestep results for the same methods. This is a good indication that

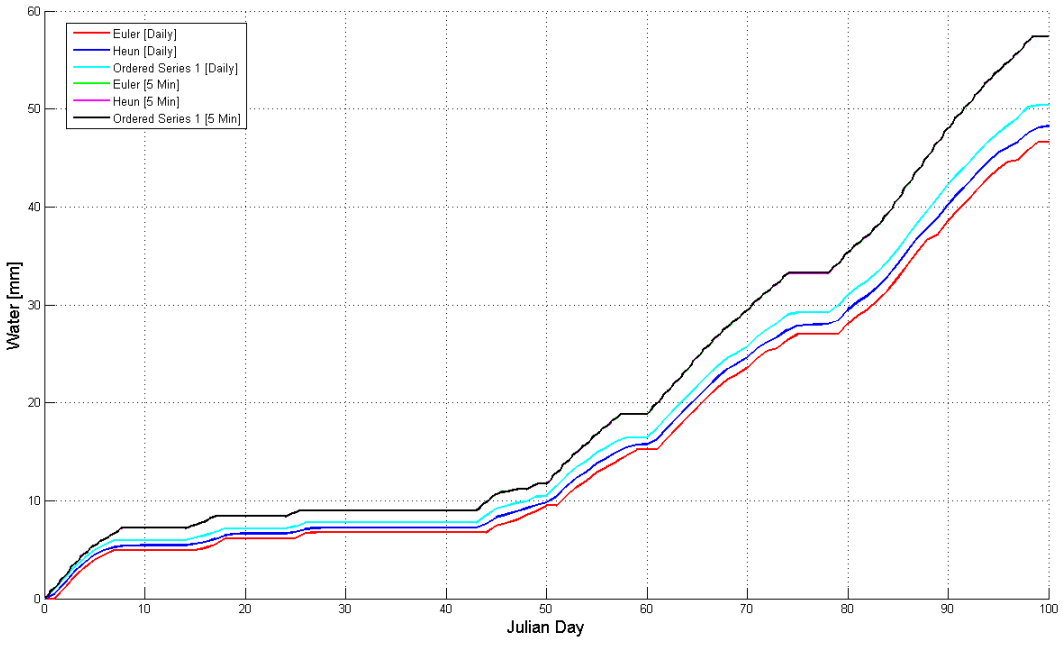

Figure 4.15: Cumulative open water evaporation from TC1 using the ordered series, Euler method and iterative Heuns method for multiple timesteps

while the results generated by various operator orders is prone to error, the Euler and Heuns methods at the daily timestep are capable of replicating truth at similar computational cost.

A fine resolution graph (figure 4.17) of the lower soil layer clearly shows the iterative Heuns methods ability to maintain a good approximation of truth during the simulation. This figure provides a straightforward example of how variations in operator order can affect the mathematical accuracy of hydrological models. Since operation order does not affect the Euler method or the iterative Heuns method, they are superior numerical methods for use in hydrological models. However, the likelihood of the Euler method to violate thresholds makes it behave worse than the ordered series approach.

4.6.3 Error Analysis

Understanding of the three numerical methods used during Ravens simulation of the Nith River watershed requires knowledge of the errors inherent to the solvers. This section discusses the absolute associated with each method, as well as the

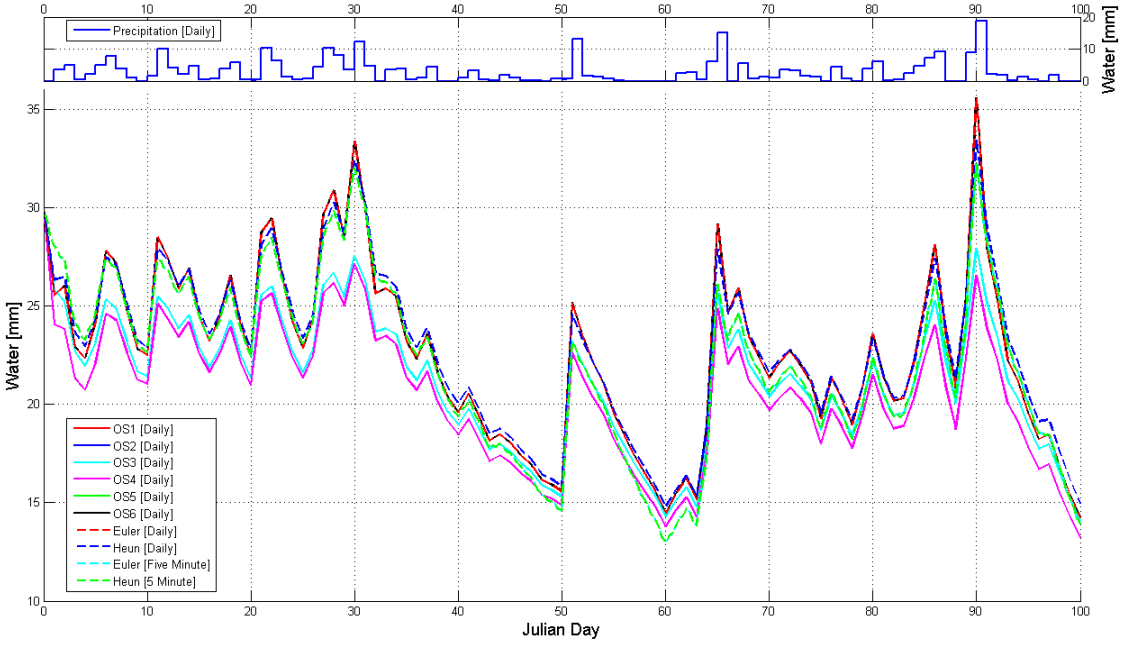

Figure 4.16: Upper soil layer storage from TC2 using ordered series, Euler and iterative Heuns method for multiple timesteps

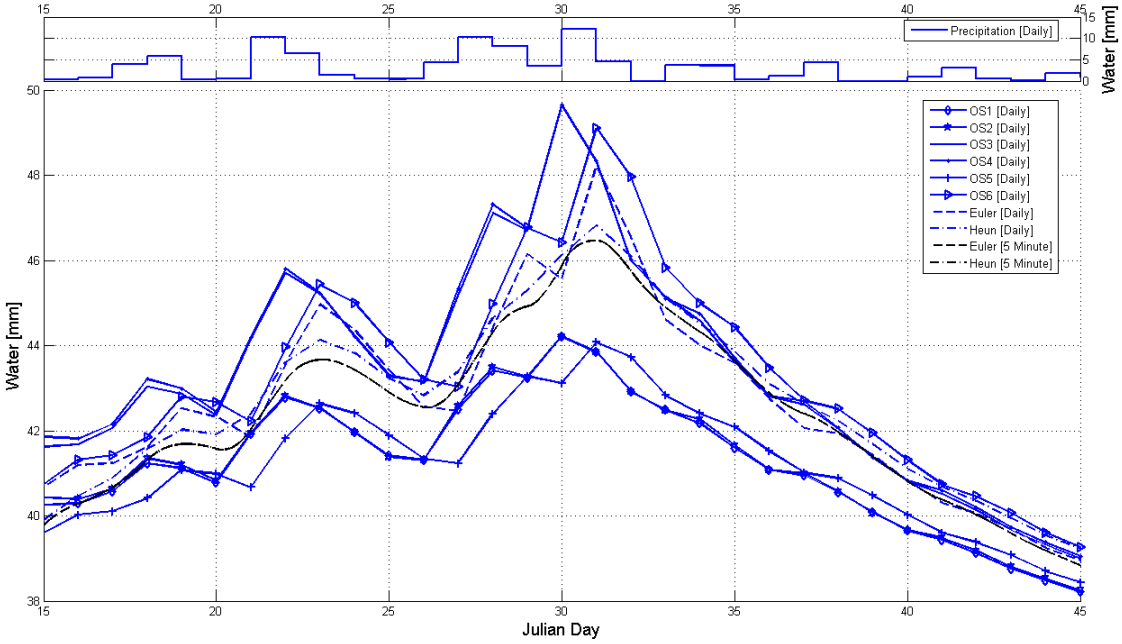

Figure 4.17: Truncated lower soil layer storage from modified simulation using ordered series, Euler and iterative Heuns method for multiple timesteps

error associated with timestep size, during their use in TC2. Absolute error was calculated using:

$$\xi_{Abs} = \left\| x - x_a \right\| \tag{4.1}$$

where ξ_{Abs} is the absolute error, x is the 'exact' solution for a given diagnostic variable (here represented by the 5 minute timestep), and x_a is the approximation of the solution. Relative error was also computed but is not included in these results since it provided little additional information to the absolute error.

The ordered series approach, which is assumed to contain significant errors, shows that as timestep size is decreased, the absolute error decreases. As timestep size shrinks from daily to 11.25 minutes, the error is reduced by a factor of 100.

The absolute error associated with the Euler method is generally slightly lower than the error produced by the ordered series approach, although only marginally. The error decreases as timestep size shrinks from the daily to the 11.25 minute and reduces by the same magnitude as the ordered series approach .

The iterative Heuns method absolute error (figure 4.18) is lower than both the ordered series and Euler methods. The results show that at a daily timestep, the error associated with the iterative Heuns method is less than 1. This raises confidence that the iterative Heuns method should have increased mathematical accuracy than the other methods implemented at a daily timestep.

An assessment of the absolute error at every timestep was conducted to see how error changes as timestep size decreases. Figure 4.19 shows that for each numerical method implemented, as the timestep size decreases, the error also decreases. As expected, the error associated with the ordered series approaches varies depending on the operation order and timestep. The error does decrease, but not in a consistent manner for all orders. The iterative Heuns method graph shows that the error decreases with timestep size in an almost consistent fashion. An interesting trait occurs in all methods: the 12 hour timestep shows larger errors than all other timesteps including the daily timestep size. The author is unable to explain this anomaly.

4.6.4 Numerical Method Comparison Summary

The numerical method that most accurately represents truth is unfortunately not easy to pinpoint from the tests done in this section. Tests performed using TC1

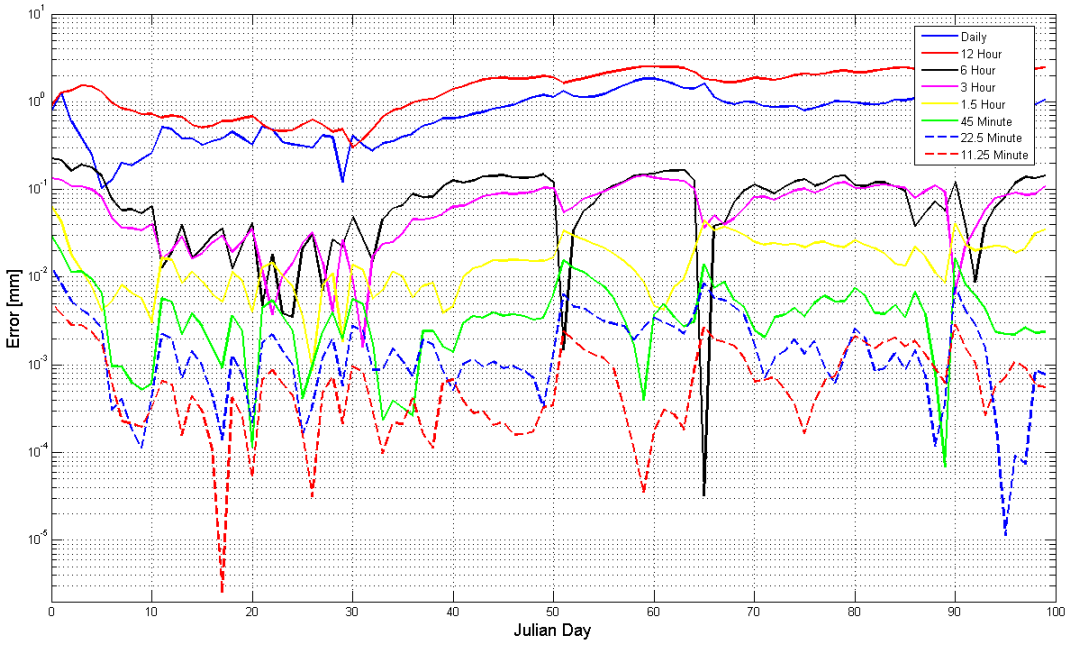

Figure 4.18: Absolute error associated with the iterative Heun method in the upper soil layer from TC2 at multiple timesteps

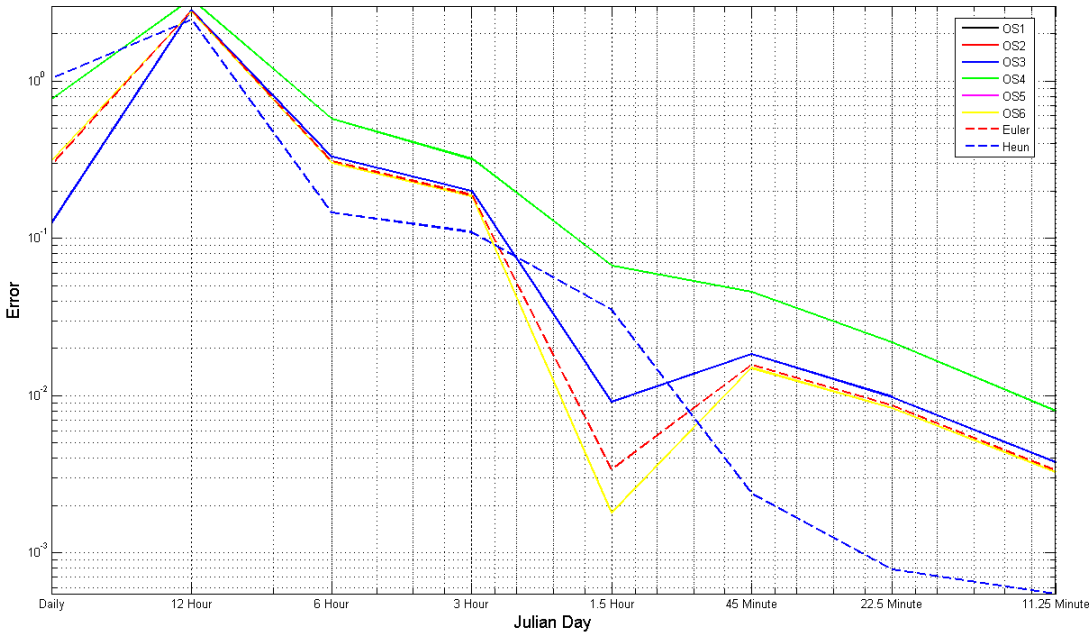

Figure 4.19: Absolute error associated with all numerical methods and ordered series approaches in the upper soil layer from TC2 at multiple timesteps

tend to show that the ordered series approach is the best approximation of truth when analyzing the amount of water in storage units over time. However, analyses of cumulative redistribution fluxes show that the Euler method is also capable of closely approximating truth. When using TC2, it is clear that the iterative Heun method is the best approximation of truth. Unlike the ordered series methods which varies depending on the operator order and the Euler method which is a near approximation of truth, the iterative Heun method maintains a consistently close approximation of truth. It is likely that the modification of existing models to incorporate high order Runge-Kutta methods will benefit hydrological models. The most important reason for this is the fact that it negates the impact of operation order when calculating redistribution rates in hydrological models.

4.7 Implications of Results

The results shown in the previous sections demonstrate that, regardless of numerical method and timestep, Raven is capable of approximating the streamflow present in the Nith River. However, as previously established, a reasonable hydrograph is not indicative of a properly calibrated or necessarily effective model [Kirchner, 2006].

For test case 2, simulations using the ordered series approaches show that the solution approximation is dependent upon the operator order, though the degree of dependence was less than might be expected. The conventional ordered series approach has been shown to be susceptible to some operator splitting errors that may lead to discrepancies in model predictions of water/ energy storage values and redistribution fluxes of a watershed. The minor operator splitting errors found in these results are likely due to this particular test case. Alternate test cases may produce larger and therefore more worrisome errors. The implications of this is that many existing hydrological models could produce diagnostic results that contain significant errors. These errors can affect predictions and policy or management decisions that are based upon those predictions. These errors cannot be discerned without the ability to either (1) more finely resolve the timestep or (2) run the model with a more robust algorithm. Understanding how the ordered series method operates in practise is essential to improving the numerical methods used in existing models and distributed hydrological models overall.

The iterative Heuns method, at a daily timestep, appears to be slightly more accurate than the Euler and ordered series methods, at similar computational cost.

The use of the iterative Heuns method in distributed hydrological models is beneficial for mathematical accuracy. Higher order numerical methods can increase the runtime of a model, however being able to produce results at a daily timestep that are equivalent to the results at small timesteps makes the iterative Heuns method a potentially valuable contribution to distributed hydrological modelling. It is important to note that within Raven (and likely other hydrological models), numerical method choice does not have a large impact on the model runtime. The bulk of runtime is spent reading and writing from input/output files.

The sensitivity of hydrological processes can be assessed based on the results generated while using test case 2. It can be seen that some processes are dominant within the watershed; soil evaporation and open water evaporation are fast acting processes that move large quantities of water. The other processes that define the system are constrained by threshold behaviour. All surface water processes are prone to threshold constraints which limit the redistribution of water and energy within a system. Parameter values within the simulation can act to increase or decrease the effect of individual process rates. However, the increase or decrease of a few parameters (e.g., soil thickness, soil evaporation PET fraction, maximum depression storage) has the ability to control and significantly modify the results obtained from a model. It is assumed that arid systems would show more sensitivity to parameters and threshold behaviour than continuously saturated areas. This is because arid areas have a finite amount of moisture which would promote competition between hydrological processes.

Modellers, when designing future hydrologic models, should consider the potential for inaccuracy based on numerical method choice. The traditional solution method, the ordered series approach, should be implemented with knowledge of the inherent errors associated with its use. The implementation of alternative numerical methods, such as the iterative Heuns method or other high order implicit methods, can increase model accuracy, and can be done without significant increases in computational cost.

Chapter 5

Conclusions

The results of the simulations run with Raven using three numerical methods and multiple timesteps show that many hydrological models have the potential to generate errors when replicating the internal processes occurring in a watershed. Through simulations using the ordered series approach with multiple operation orders, it is clear that results cannot be predicted with ease and perform in unexpected ways.

The results produced by Raven show that variations in operator order can change how the internal storage and water fluxes in a watershed are determined. This has been overlooked for many years, in part because the typical method for verifying models has been to use streamflow gauges for calibration. The results clearly show that hydrographs produced using multiple operator orders are unaffected by the sequence order. It has been shown, through the use of ordered series approaches, that hydrographs can be replicated with a high degree of confidence while the other diagnostics, such as snow depth or soil moisture may vary. This reinforces the belief that operator splitting errors can increase during the use of ordered series methods even though TC1 and TC2 resulted in less significant errors that hypothesized. Alternate test cases may show larger operator splitting errors when tested.

The use of higher order Runge-Kutta methods, such as the Euler method and the iterative Heuns method, produce results that can be more accurate using a daily timestep than results produced by the ordered series approach at the same timestep. If Runge-Kutta methods are incorporated into existing models, it would lead to 2 improvements; (1) operator order would cease to be an issue in the operation of models and (2) models, using a daily timestep, may produce results that are comparable to results achieved at smaller timesteps using lower order numerical

methods. As these results have shown, variations in timestep size appear to have a greater effect on the accuracy of hydrological model results then the choice of numerical method.

The importance of this research is to show that many existing models may produce results that cannot be confidently relied upon. This is not just due to the selection of subprocess algorithms or parameters, but also the numerical method. The understanding of the inner workings of a watershed model are necessary to properly simulate storage unit interactions. This thesis shows that the internal storage and redistribution rates are more sensitive than outflow hydrographs to timestep size and numerical method choice. The numerical methods and algorithms implemented in Raven are transferable and can be implemented in existing models. Raven currently has the ability to emulate many existing models and the numerical methods tested have been used within those emulations.

This research can be expanded to include studies into the effects of ordered series approaches when implemented in other types of complex distributed models (e.g., for climate prediction or contaminant transport). Additional numerical methods can also be implemented and tested to see if they produce more accurate results.

References

Albert, M. and G. Krajeski (1998). A fast, physically based point snowmelt model for use in distributed applications. *Hydrological Processes*, 1809–1824.

Anderson, E. (1976). A point energy and mass balance model of a snow cover. *NOAA Technical Report NWS 19*.

ASCE (1969). *Design and construction of sanitary and storm sewers*. New York: Manual Reports No. 9, ASCE.

Bathurst, J. and P. O'Connell (1992). Future of distributed modelling: The systeme hydrologique europeen. *Hydrological Processes 6*, 265–277.

Beven, K. (1997a). *Distributed Modelling in Hydrology: Applications of the TOP-MODEL concepts*. Chichester: John Wiley & Sons Ltd. 9

Beven, K. (1997b). Topmodel: A critique. *Hydrological Processes 11*, 1069–1085. 9, 12, 74

Beven, K. (2001). *Rainfall-Runoff Modelling: The Primer*. Chichester: John Wiley & Sons Ltd. 25

Bingeman, A., N. Kouwen, and E. Soulis (2006). Validation of the hydrological processes in a hydrological model. *Journal of Hydrologic Engineering 11*, 451–463. 10

Booch, G. (1994). *Object-Oriented Analysis and Design with Applications (2nd Edition)*. Redwood City, CA: Benjamin/ Cummings Publisher Co. 21

Borah, D., J. Arnold, M. Bera, E. Krug, and X. Liang (2007). Storm event and continuous hydrologic modelling for comprehensive and efficient watershed simulations. *Journal of Hydrologic Engineering 6*, 605–616. 7

Borah, D. and M. Bera (2003). Watershed-scale hydrologic and nonpoint-source pollution models: Review of mathematical bases. *Transactions of the ASAE 46*, 1553–1566. 7, 8, 9, 11, 12

Borah, D. and M. Bera (2004). Watershed-scale hydrologic and nonpoint-source pollution models: Review of applications. *Transactions of the ASAE 47*, 789–803. 8, 9, 12

Brubaker, K., A. Rango, and W. Kustas (1996). Incoporating radiation inputs into the snowmelt runoff model. *Hydrological Processes 10*, 1329–1343. 26

Carrayrou, J., R. Mosé, and P. Behra (2004). Operator-splitting procedures for reactive transport and comparison of mass balance errors. *Journal of Contaminant Hydrology 68*, 239–268. 2, 17

Chow, V., D. Maidment, and L. Mays (1988). *Applied Hydrology*. McGraw-Hill, Incorporated. 7

Clark, M., A. Slater, D. Rupp, R. Woods, J. Vrugt, H. Gupta, T. Wagener, and L. Hay (2008). Framework for understanding structural errors (fuse): A modular framework to diagnose differences between hydrological models. *Water Resources Research 44*, W00B02. 11, 22, 25, 26, 27, 28, 74

Craig, J. and A. Snowdon (2010). Raven: A rigorously formalized modular hydrological model. *Environmental Modeling and Software [Preprint submitted]*. 2, 20, 21

Cranmer, A., N. Kouwen, and S. Mousavi (2001). Proving watflood: modelling the nonlinearities of hydrologic response to storm intensities. *Canadian Journal of Civil Engineering 28*, 837–855.

Crawford, N. and R. Linsley (1966). *Digital simulation in hydrology: Stanford watershed model IV*. Stanford University, Department of Civil Engineering. 5

Demaria, E., B. Nijssen, and T. Wagener (2007). Monte carlo sensitivity analysis of land surface parameters using the variable infiltration capacity model. *Journal of Geophysical Research 112*. 26

Dingman, S. (2002). *Physical Hydrology* (2nd ed.). Prentice-Hall, Incorporated. 6, 7, 8, 26, 27, 28, 74

Easton, Z., D. Fuka, M. Walter, D. Cowan, E. Schneiderman, and T. Steenhuis (2008). Re-conceptualizing the soil and water assessment tool (swat) model to predict runoff from variable source areas. *Journal of Hydrology 348*, 279–291. 8, 9

Faragó, I. (2006). Operator splittings and numerical methods. *International Conference on Large-Scale Scientific Computing 3743*, 347–354. 15

Faragó, I. (2008). A modified iterated operator splitting method. *Applied mathematical modelling 32*, 1542–1551. 15

Faragó, I. and A. Havasi (2001). The mathematical background of operator splitting and the effect of non-commutativity. *Lecture notes in computer science*, 264–271. 15

Hoffman, J. D. (1992). *Numerical Methods for Engineers and Scientists*. McGraw-Hill, Inc. 13, 14, 17, 18, 36

Jacques, D., J. Šimůnek, D. Mallants, and M. van Genuchten (2006). Operator-splitting errors in coupled reactive transport codes for transient variably saturated flow and contaminant transport in layered soil profiles. *Journal of Contaminant Hydrology 88*, 197–218. 2, 16

Kanney, J., C. Miller, and D. Barry (2003). Comparison of fully coupled approaches for approximating nonlinear transport and reaction problems. *Advances in Water Resources 26*, 353–372. 16

Kanney, J., C. Miller, and C. Kelley (2003). Convergence of iterative split-operator approaches for approximating nonlinear reactive transport problems. *Advances in Water Resources 26*, 247–261. 2, 15, 17

Kavetski, D. and G. Kuczera (2007). Model smoothing strategies to remove microscale discontinuities and spurious secondary optima in objective functions in hydrological calibration. *Water Resources Research 43*, W03411. 13

Kavetski, D., G. Kuczera, and S. Franks (2003). Semidistributed hydrological modeling: A 'saturation path' perspective on topmodel and vic. *Water Resources Research 39*, 1246. 10, 11, 74

Kavetski, D., G. Kuczera, and S. Franks (2006). Calibration of conceptual hydrological models revisited: 1. overcoming numerical artefacts. *Journal of Hydrology 320*, 173–186. 12

Khan, S., L. Yufeng, and A. Ahmad (2007). Analysing complex behaviour of hydrological systems through a systems dynamic approach. *Environmental Modelling and Software*, 1–10.

Kirchner, J. (2006). Getting the right answers for the right reasons: Linking measurements, analyses, and models to advance the science of hydrology. *Water Resources Research 42*. 62

Kouwen, N., M. Danard, A. Bingeman, W. Luo, F. Seglenieks, and E. Soulis (2005). Case study: Watershed modeling with distributed weather model data. *Journal of Hydrologic Engineering 10*, 23–38. 10

Kustas, W., A. Rango, and R. Uijlenhoet (1994). A simple energy budget algorithm for the snowmelt runoff model. *Water Resources Research 30*, 1515–1527. 26

Lang, G. (2008). Nith river: Land use. `http://info.wlu.ca/~wwwgeog/special/grand/f1_lu.htm`. viii, x, 38, 39, 41

Lee, H., E. Zehe, and M. Sivapalan (2007). Predictions of rainfall-runoff response and soil moisture dynamics in a microscale catchment using the crew model. *Hydrology and Earth System Science 11*, 819–849. 6

Lemonsu, A., V. Masson, and E. Berthier (2007). Improvement of the hydrological component of an urban soil-vegetation-atmosphere-transfer model. *Hydrological Processes 21*, 2100–2111.

Liang, X., D. Lettenmaier, E. Wood, and S. Burges (1994). A simple hydrologically based model of land surface water and energy fluxes for general circulation models. *Journal of Geophysical Research 99*, 14415–14428. 26

Liang, X., E. Wood, and D. Lettenmaier (1996). Surface soil moisture parameterization of the vic-2l model: Evaluation and modification. *Global and Planetary Change 13*, 195–206. 27

Liang, X., Z. Xie, and M. Huang (2003). A new parameterization for surface and groundwater interactions and its impact on water budgets with the variable infiltration capacity (vic) land surface model. *Journal of Geophysical Research 108*. 10, 12

Linsley, R., M. Kohler, and J. Paulhus (1949). *Applied Hydrology*. New York: McGraw-Hill. 74

Maidment, D. (1992). *Handbook of Hydrology*. McGraw-Hill, Incorporated. 74

Matott, L. (2005). Ostrich: An optimization software tool; documentation and users guide, version 1.6. `http://www.groundwater.buffalo.edu`. 42

McKillop, R., N. Kouwen, and E. Soulis (1999). Modeling the rainfall-runoff response of a headwater wetland. *Water Resources Research 35*, 1165–1177.

Phillip, J. (1954). An infiltration equation with physical significance. *Soil Science 77*, 153–157.

Pomeroy, J. (2008). personal communication. 17

Pomeroy, J., D. Gray, T. Brown, N. Hedstrom, W. Quinton, R. Granger, and S. Carey (2007). The cold regions hydrological model: a platform for basing process representation and model structure on physical evidence. *Hydrological Processes 21*, 2650–2667. 10

Presant, E. and R. Wicklund (1971). *The Soils of Waterloo County, Report No. 44 of the Ontario Soil Survey*. Department of Soil Science, University of Guelph and Ontario Department of Agriculture and Food. 38, 41

Refsgaard, J. and J. Knudsen (1996). Operational validation and intercomparison of different types of hydrological models. *Water Resources Research 32*, 2189–2202.

Seglenieks, F. (2008). Uw weather station: Data archives. `http://www.weather.uwaterloo.ca/`. 41

Simpson, M. and K. Landman (2008). Theoretical analysis and physical interpretation of temporal truncation errors in operator split algorithms. *Mathematics and Computers in Simulation 77*, 9–21. 2, 17

Singh, V. and D. Frevert (2006). *Watershed Models*. CRC Taylor & Francis. 7

Singh, V. and D. Woolhiser (2002). Mathematical modeling of watershed hydrology. *Journal of Hydrologic Engineering 7*, 270–303.

Soulis, E. (2009). personal communication. 10

Tetzlaff, D., J. McDonnell, S. Uhlenbrook, K. McGuire, P. Bogaart, F. Naef, A. Baird, S. Dunn, and C. Soulsby (2008). Conceptualizing catchment processes: simply too complex? *Hydrological Processes 22*, 1727–1730.

Todini, E. (1988). Il modello afflussi deflussi del fiume arno. *Relazione Generale dello studio per conto della Regione Toscana [Technical Report]*. 10

Todini, E. (1996). The arno rainfall-runoff model. *Journal of Hydrology 175*, 339–382. 10, 27

USGS (2008). The water cycle. `http://ga.water.usgs.gov/edu/watercycle.html`. viii, 6

Valocchi, A. and M. Malmstead (1992). Accuracy of operator splitting for advection-dispersion-reaction problems. *Water Resources Research 28*, 1471–1476. 2, 16

Verseghy, D. (1991). Class - a canadian land surface scheme for gcms. i. soil model. *International Journal of Climatology 11*, 111–133. 15

Wang, A., K. Li, and D. Lettenmaier (2008). Integration of the variable infiltration capactiy model soil hydrology scheme into the community land model. *Journal of Geophysical Research 113*, 1–15.

Wang, Z., O. Batelaan, and F. D. Smedt (1997). A distributed model for water and energy transfer between soil, plants and atmosphere (wetspa). *Physics and Chemistry of the Earth 21*, 189–193.

Welsh, W. (2008). Water balance modelling in bowen, queensland, and the ten iterative steps in model development and evaluation. *Environmental Modelling and Software 23*, 195–205.

Wood, E., D. Lettenmaier, and V. Zartarian (1992). A land-surface hydrology parameterization with subgrid variability for general circulation models. *Journal of Geophysical Research 97*, 2717–2728. 25

Yuan, F., Z. Xie, Q. Liu, H. Yang, F. Su, X. Liang, and L. Ren (2004). An application of the vic-3l land surface model and remote sensing data in simulating streamflow for the hanjiang river basin. *Canadian Journal of Remote Sensing 30*, 680–690. 10, 12

Zanotti, F., S. Endrizzi, G. Bertoldi, and R. Rigon (2004). The geotop snow module. *Hydrological Processes 18*, 3667–3679.

Zehe, E. and M. Sivapalan (2009). Threshold behaviour in hydrological systems as (human) geo-ecosystems: manifestations, controls, impliations. *Hydrology and Earth System Sciences 13*, 1273–1297.

Zhang, X., G. Hormann, and N. Fohrer (2008). An inverstigation of the effects of model structure on model performance to reduce discharge simulation uncertainty in two catchments. *Advances in Geoscience 18*, 31–35. 14

APPENDICES

Appendix A

Appendix A: Hydrological Processes available in Raven

A.1 Hydrological Processes

The following tables summarize the hydrological processes that are currently available in the distributed hydrological model Raven.

[1]From Dingman [2002]
[2]From Maidment [1992]
[3]From Clark et al. [2008]
[4]From Beven [1997b]
[5]From Kavetski et al. [2003]
[6]From Linsley et al. [1949]

Process	**Method**	**Equation**
Potential Evapotranspiration		
	Penman-Combination[1]	$PET = \frac{m_{et} \cdot (K+L) + \gamma \cdot e_v \cdot \rho_w \cdot \lambda_v \cdot v \cdot e_{sat} \cdot (1-W_a)}{\rho_w \cdot \lambda_v \cdot (m_{et}+\gamma)}$
	Penman-Monteith[1]	$PET = \frac{m_{et} \cdot (K+L) + \rho_a \cdot c_a \cdot C_{at} \cdot e_{sat} \cdot (1-W_a)}{\rho_w \cdot \lambda_v \cdot [m_{et} + \gamma \cdot (1 + \frac{C_{at}}{C_{can}}]}$
	Hargreaves[2]	$PET = 0.0075 R_a C_t \delta_t^{\frac{1}{2}} T_{avg\delta}$
	Priestley-Taylor[1]	$PET = \frac{1}{\lambda} \frac{m_{et}(R_n - G)}{s+\gamma} \alpha_{PT}$
	Constant	User specified rate
Soil Evaporation		
Moves water	VIC[3]	$\frac{\partial S}{\partial t}\Big\vert_{SE} = PET\left(1 - (1 - \frac{S_{sat}}{S_{\max}})^{\gamma_V}\right)$
from uppermost	TOPMODEL[3]	$\frac{\partial S}{\partial t}\Big\vert_{SE} = PET \frac{\min(S_1^T, S_{1,\max}^T)}{S_{1,\max}^T}$
soil layer(s)	Sequential[3]	$\frac{\partial S_1}{\partial t}\Big\vert_{SE} = PET \frac{\min(S_1^T, S_{1,\max}^T)}{S_{1,\max}^T}$
to atmosphere		$\frac{\partial S_2}{\partial t}\Big\vert_{SE} = \left(PET - \frac{\partial S_1}{\partial t}\right) \frac{\min(S_2^T, S_{2,\max}^T)}{S_{2,\max}^T}$
through	Root Weighting[3]	$\frac{\partial S_1}{\partial t}\Big\vert_{SE} = PET r_1 \frac{\min(S_1^T, S_{1,\max}^T)}{S_{1,\max}^T}$
evapotranspiration (ET)		$\frac{\partial S_2}{\partial t}\Big\vert_{SE} = PET r_2 \frac{\min(S_2^T, S_{2,\max}^T)}{S_{2,\max}^T}$
Canopy Evaporation		
moves water from canopy	Constant	$\frac{\partial S}{\partial t}\Big\vert_{CE} = E_c F_c$
to atmosphere via ET	Rutter[1]	$\frac{\partial S}{\partial t}\Big\vert_{CE} = (1 - F_t) PET \frac{S_0}{C_{cap} F_c}$
Canopy Drip/ Drainage		
moves water from canopy	Slow Drain	$\frac{\partial S}{\partial t}\Big\vert_{Drip} = S_0 - F_c C_{cap} + C_D \frac{S_0}{F_c}$
to surface, ponded,	Rutter[1]	$\frac{\partial S}{\partial t}\Big\vert_{Drip} = (1 - P_p) M_C$
and soil storage		$\frac{\partial S}{\partial t}\Big\vert_{Drain} = \frac{\partial (C_{stor} - M_D)}{\partial t}$

Table A.1: Evaporation and Canopy processes currently incorporated in Raven

Process	Method	Equation
Percolation		
moves water from soil layer to lower soil layer	VIC/ TOPMODEL FUSE[3]	$\frac{\partial S}{\partial t}\Big\|_{Perc} = k_u \left(\frac{S_1}{S_{1,\max}}\right)^c$
Baseflow		
moves water from soil and ground storage to surface water storage	Bucket	$\frac{\partial S}{\partial t}\Big\|_{Base} = \alpha_b S_0$
	VIC FUSE[3]	$\frac{\partial S}{\partial t}\Big\|_{Base} = k_s \left(\frac{S_2}{S_{2,\max}}\right)^n$
	TOPMODEL FUSE[3]	$\frac{\partial S}{\partial t}\Big\|_{Base} = \frac{k_s m}{\lambda_n^n}\left(\frac{S_2}{mn}\right)^n$
Quickflow		
	VIC SPM[5]	$\frac{\partial S}{\partial t}\Big\|_{Qf} = P_p \left(\frac{(S_{\max} - S_0)}{(S_{\max} - S_{crit})}\right)^{\frac{1}{\alpha_V}+1}$

Table A.2: Soil processes currently incorporated in Raven

Process	Method	Equation
Snowmelt		
moves water from snow to soil and surface storage	Degree Day[1]	$\frac{\partial S}{\partial t}\Big\|_{Melt} = \min(S_0, M_a(T_a - T_f))$
	Hybrid[1]	$\frac{\partial S}{\partial t}\Big\|_{Melt} = \min\left[\left(\frac{K+L}{\rho_w \lambda_f}\right) + M_r T_a, S_0\right]$
Sublimation		
moves water from snow to atmosphere	Kuzmin[6]	$\frac{\partial S}{\partial t}\Big\|_{Sub} = 0.18 + 0.098 v(e_{sat} - e)$
	Sverdrup[6]	$\frac{\partial S}{\partial t}\Big\|_{Sub} = \frac{0.623 \rho_a k^2 v(e_{sat} - e)}{P_a (log(\frac{800}{R_s}))^2}$
	Central Sierra[6]	$\frac{\partial S}{\partial t}\Big\|_{Sub} = 0.0063 (v_{ht} e_{ht})^{-\frac{1}{6}} (e_{sat} - e) v$
	Williams[6]	$\frac{\partial S}{\partial t}\Big\|_{Sub} = 0.00011 \rho_a v(e - e_{sat})$

Table A.3: Snow processes currently incorporated in Raven

Process	**Method**	**Equation**
Runoff		
	Partition[1]	$\frac{\partial S}{\partial t}\Big\|_{Run} = (1 - P_c)P_p$
	Mass Balance[1]	$\frac{\partial S}{\partial t}\Big\|_{Run} = 0.0$
	SCS Method[2]	$\frac{\partial S}{\partial t}\Big\|_{Run} = \frac{(P_p - 0.2r)^2}{(P_p + 0.8r)}$
	Rational[2]	$\frac{\partial S}{\partial t}\Big\|_{Run} = c_r R I_r A$
	Green-Ampt[1]	$\frac{\partial S}{\partial t}\Big\|_{Run} = P_p - \left[K_e\left(1 + \frac{(\psi - \Phi_i)S_f}{F_i}\right)\right]$
	VIC[1]	$\frac{\partial S}{\partial t}\Big\|_{Run} = P_p(1 - K_1(S_{at} - S_{\max})^{\gamma_{V2}})$
	VIC FUSE[3]	$\frac{\partial S}{\partial t}\Big\|_{Run} = P_p A_{sat}$
	TOPMODEL FUSE[3]	$\frac{\partial S}{\partial t}\Big\|_{Run} = P_p A_{sat}$
Routing		
	Muskingum[6]	$Q_o^{j+1} = C_1 I^{j+1} + C_2 I^j + C_3 Q_o^j$
	Storage Coefficient	$Q_o = \frac{1}{(t_t + 0.5)} vol$
	Mannings[6]	$Q_o = \frac{A_{ch}}{M_n}\left(\frac{A_{ch}}{P_{ch}}\right)^{\frac{2}{3}}\sqrt{R_m}$
	None	$Q_o = 0.0$

Table A.4: Overland processes currently incorporated in Raven