QUANTIFYING THE EFFECTS OF SUSTAINABLE INTENSIFICATION PRACTICES ON STREAMFLOW AND WILDLIFE: A CASE STUDY IN LAIKIPIA, KENYA

By

LORY LEE WILLARD

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2023

To my parents

ACKNOWLEDGMENTS

I thank the University of Florida Agricultural and Biological Engineering Department for a Graduate Fellowship that supported me while I completed my studies, and The Nature Conservancy (Award AWD07062) for research funding and partnership.

I extend my gratitude to Dr. Rafael Muñoz-Carpena and Dr. Cheryl Palm who have provided guidance and mentorship for both research and life during these past few years. I also thank the members of my committee, Drs. Greg Kiker, Gabriel Maltais-Landry, Kelly Caylor, Mike Annable, and Young Gu Her for their guidance and support. I am eternally grateful to John Gitonga and his attention to detail as well as sense of humor in the field, and thankful to have found a great friend in him during this process. In addition, I would like to thank our colleagues at The Nature Conservancy, CETRAD, Mpala Research Centre, ICRAF, Cropnuts, KALRO, and the Ewaso Ng'iro Water Resources Administration, and all of the wonderful farmers we were able to work with for supporting our work in Laikipia and providing meaningful feedback and guidance. I would also like to thank all of the UF ABE Department staff for their support and smiles that help keep all the grad students moving forward.

Finally, I am grateful to all of the family and friends who supported me through this time. Thank you to my lifelong friends who have supported me from afar and all of the amazing friends I have met in Gainesville to make it feel like home. Thank you to my mom and dad for always being there.

TABLE OF CONTENTS

page

CHAPTER

APPENDIX

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AfSis	Africa Soil Information Service
CETRAD	Centre for Training and Integrated Research in Arid and Semi-Arid Land Development
CFSR	Climate Forecast System Reanalysis
CN	Curve Number
GA	Green-Ampt
ISRIC	International Soil Reference and Information Centre
LST	Land Surface Temperature
MRC	Mpala Research Centre
NASA	National Aeronautics and Space Administration
NGO	Non-governmental organization
NSE	Nash-Sutcliffe Efficiency
SAI	Sustainable agricultural intensification
SMAP	Soil Moisture Active Passive

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

QUANTIFYING THE EFFECTS OF SUSTAINABLE INTENSIFICATION PRACTICES ON STREAMFLOW AND WILDLIFE: A CASE STUDY IN LAIKIPIA, KENYA

By

Lory Lee Willard

December 2023

Chair: Rafael Muñoz-Carpena
Major: Agricultural and Biological Engineering

Sustainable agricultural intensification (SAI) aims to produce more food per land unit while preserving ecosystem functions in food insecure regions. SAI scenarios have been tested primarily on large, homogenous agricultural lands using monthly models with high data requirements. These models are not necessarily sensitive to SAI scenarios and are temporally too coarse to evaluate ecosystem impacts. As Kenya increases irrigated agriculture, understanding impacts on savanna regions, including water availability for wildlife, is critical. This study focuses on two SAI practices, reduced tillage (RT) and drip irrigation (DI), to determine how practice adoption along the steep rainfall gradient in the Mount Kenya region impacts hydrology. We hypothesize that, along the rainfall gradient and compared to conventional agriculture, RT is more effective in dryland areas and DI is preferable in high rainfall areas for maintaining downstream flows. We monitored streamflow and rainfall with 8 streamgages and 5 rain gauges from 2021 to 2023. Kenya is relatively data-scarce with few rigorously-validated high-resolution remote sensing products. Therefore, a comparatively simple process model was developed and partially calibrated using streamflow data. The CUENCA link and node model is a daily, semi-distributed hydrologic model developed with minimal, flexible data requirements that is sensitive to tillage and irrigation. Users can simulate rainfall-runoff response with Curve

Number (CN) or Green-Ampt (GA) methods. Here, the uncalibrated CN model was used to evaluate scenarios of RT, DI, and conventional agriculture counterparts. Results indicate that high rainfall DI reduces flow volume up to 60% locally, but watershed impacts are negligible. RT may reduce local flow up to 21%, but at the savanna scale this translates to a reduction of only 5% and potential modest baseflow recovery. For wildlife requiring frequent water (e.g. Grevy’s zebra), implementing intermediate rainfall DI results in 23 more consecutive dry days than high rainfall DI. All RT scenarios performed similarly, but implementation in intermediate rainfall areas may preferentially balance flow reduction with baseflow recovery. These scenarios should be evaluated using the GA method and field-observed soil properties with a calibrated model. Results can inform sustainable water and agricultural management within Laikipia and regions under fast development.

CHAPTER 1
INTRODUCTION

1.1 Conservation Tillage and Conservation Agriculture

Sustainable agricultural intensification (SAI) has frequently been cited as a way to balance the increased food production needs of society with protection of natural resources. However, these claims have rarely been quantified using a combination of field scale data and corresponding site-specific models, even though positive impacts of conversion to SAI are highly dependent on local soils and climate (Blanco-Canqui and Ruis, 2018).

Conservation tillage systems, an important component of SAI, have been promoted and studied since the 1930s Dust Bowl in the midwestern United States (Lal, 2001). These systems preserve soil health by reducing runoff and erosion and reversing loss of soil organic matter via a minimum 30% permanent soil coverage through residue retention (Hobbs et al., 2008). The term conservation tillage was adopted as an umbrella term to include no-tillage, minimum (reduced)-tillage, direct-drilling, and any other practice that reduced soil disturbance toto conserve soil characteristics such as moisture, structure, nutrients, and biota and to conserve farmer investment such as labor and fuel (Baker et al., 2002). The Food and Agriculture Organization (FAO) adopted the term ‘conservation agriculture’ (CA) to encompass the combination of no-tillage, at least 30% permanent soil cover through residues or cover crops, and crop rotations of at least three crops (FAO, 2020b). This term has caused some confusion among farmers and academics, since conservation tillage includes some of the tenets of conservation agriculture, but it typically allows for more soil disturbance (Hobbs et al., 2008).

The majority of conservation tillage adoption, and by extension conservation agriculture, has occurred in the US, Brazil, Argentina, Canada, Australia, and China (Derpsch, 2008; Kassam et al., 2014). The conversion to conservation agriculture has provided some quick relief to erosion and runoff problems in these countries, but total area under cultivation continues to expand around the globe and especially in developing countries (Baveye et al., 2011). While farmers that currently practice CA are large-scale highly mechanized farms (Kassam et al., 2010), some argue that these practices can also be adopted by smallholder farmers in Sub Saharan Africa and Asia (Wall, 2007). Smallholder systems are often mixed crop-livestock systems and the farmers face challenges due to limited resources, including land, labor, and capital, that may ultimately mean that CA is not feasible (Valbuena et al., 2012).

Benefits of reduced-tillage (including no-till) and residue retention, both components of CA, are well-documented at the field scale in the US and South America. Verhulst et al. (2010) summarizes general changes to soil physical properties (e.g. aggregate stability, bulk density, porosity, and infiltration), chemical properties (e.g. nutrients, minerals, CEC, and pH), and biological properties (e.g. organic matter in topsoil, microbial biomass, earthworm presence, and arthropod diversity) after adoption of no-tillage and residue retention, and found that most properties improve. Conducting a meta-analysis of no-tillage and reduced-tillage effects on soil physical properties, Blanco-Canqui and Ruis (2018) found improvements in compactibility (i.e. reduced compactibility), wet aggregate stability, water infiltration and available water, but mixed results for no-till effects on bulk density and saturated hydraulic conductivity. They concluded that during rainfall events, no-till may have a lower risk of compaction via controlled trafficking than reduced-till, but time since no-tillage adoption and soil textural class are important factors. They also emphasize the importance of conducting field-scale studies in a wider variety of agro-ecosystems and soil types (Blanco-Canqui and Ruis, 2018).

Both field-scale and watershed-scale studies have attempted to quantify benefits of conservation tillage. Field-scale studies have indicated that conversion to conservation tillage can have mixed results reducing runoff flow and sediment loads depending on rainfall intensity, evapotranspiration, and soil type (Algoazany et al., 2007; Bosch et al., 2012; Didone et al., 2014; Endale et al., 2014), but there is evidence that soils under long-term conservation tillage hold more plant available soil water, improve deep drainage, increase baseflow, allow for quicker baseflow recovery after drought, and require less irrigation (Tomer et al., 2005; Baumhardt et al., 2017; Assefa et al., 2018). Rawls et al. (1980) developed a relationship between residue retention (as a proxy for tillage intensity) and percent curve number reduction to simulate the change in saturated hydraulic conductivity associated with reduced tillage. Saturated hydraulic conductivity does not have a consistent response to no-till across field studies and different time scales under no-till (Strudley et al., 2008), in some cases not indicating any change in properties until 10 years post-no-till adoption (Chang and Lindwall, 1992).

Conservation effects at the watershed scale have been assessed primarily through hydrological models. The Soil and Water Assessment Tool (SWAT) and Annualized Agricultural Non-Point Source (AnnAGPS) are the most common models used to assess potential benefits of conservation practices.. These were employed during the USDA-ARS Conservation Effects Assessment Project devised to quantify the effects of conservation practices across USA working lands (Tomer and Locke, 2011). Although globally used, SWAT may be better suited to predicting streamflow in humid climates (Veith et al., 2010). In addition, SWAT is more sensitive to other parameters (curve number, crop rotations, and soil coverage) than tillage practices, meaning that the impacts of changes in tillage may be underestimated when compared to scenarios of landuse change, crop rotations, or soil coverage (Ullrich and Volk, 2009). Tomer and Locke (2011) recognized the complex nature of quantifying conservation benefits at the larger watershed scale without extensive field studies. Bowmer

(2011) discusses the difficulty of attributing changes in river flow and quality in a watershed to specific agricultural practices or land use changes, with a specific discussion on scale and lag-time.

While few studies in Sub Saharan Africa focus on CA effects on soil physical properties, research in the region does exist to assess impacts of CA on crop yields (Brouder and Gomez-Macpherson, 2014; Steward et al., 2018). Globally, results on yields are mixed, but rainfed dryland systems that implement the three core practices (no-till, soil coverage, and crop rotations) seem to significantly increase productivity (Pittelkow et al., 2015). A recent meta-analysis of conservation agriculture in Eastern and Southern Africa indicates that results at the field scale are dependent on management practices as well as local factors, including rainfall and soil type (Nyagumbo et al., 2020). Depending on the crop grown and rainfall of across different African areas, irrigation may also be required for productive crop yields (Makurira et al., 2007). Overall, conservation agriculture practices reduced yield variability by 11% and performed best in well-drained (loam) soils. Crop rotations had the highest impact on yields, increasing them 35%, following by conservation tillage, which increased yields by 26%. Relative yields increased the most in areas that received less than 700 mm of rainfall per year. In areas where rainfall exceeded 1300 mm or soils were poorly drained, yields remain the same or decrease under conservation agriculture (Nyagumbo et al., 2020).

While field-scale responses of conservation tillage have been studied, there is a clear and important gap in quantifying watershed and regional streamflows and quality response to the practices across a large-scale precipitation gradients, including impacts and teleconnections on wildlife and ecology as presented later in this Chapter. This could ultimately inform managers and other stakeholders on the SAI benefits, opportunities and limitations beyond the current focus on field-scale benefits.

1.2 Irrigation and Conservation Agriculture

Since agriculture is the largest user of freshwater on the globe, there is potential to simultaneously close yield gaps and preserve environmental flows through more efficient use of irrigation water (Rockstrom and Karlberg, 2010; Mueller et al., 2012; Garnett et al., 2013). In this context, irrigation efficiency is measured as crop-productive water consumption or “crop per drop” (Seckler, 1996; Jagermeyr et al., 2015). Jagermeyr et al. (2015) estimate that improving irrigation efficiency globally through conversion to drip or sprinkler systems could reduce non-beneficial consumption losses (i.e. evaporation, interception, and runoff) by 54 -76% while maintaining production levels. However, reduction of runoff, also called “return flows” from irrigation can actually decrease river flow levels, and promotion of efficient irrigation can exacerbate over-abstraction of water (Huffaker, 2008; Ward and Pulido-Velazquez, 2008; Grafton et al., 2018). After a thorough review of drip irrigation literature, van der Kooij et al., (2013) determined that there are not consistent definitions of “efficiency” across studies and that positive impacts of drip irrigation are limited to spatial and temporal scales. The authors also address important ET factors typically left out of comparisons between drip irrigation and conventional irrigation, including lack of Kc coefficient adjustment (as recommended in FAO manual 56) (FAO, 1998), impacts of deficit irrigation on ET, and weed growth under different irrigation schemes (van der Kooij et al., 2013). Scott et al., (2013) details three watershed level examples of improved irrigation efficiency at the plot scale leading to diminishing water availability and increased salinity in surrounding ecosystems due to increased irrigation area and more water intensive crops. Pool et al., (2022) identified reduced groundwater recharge under drip scenarios compared to flood scenarios in dry years, and similar performance of systems during wet years in the Mediterranean region.

As of 2008, only 2% of cultivated land in Sub Saharan Africa was irrigated, but this land accounted for 20% of food production (Foster and Briceno-Garmendia, 2009). To meet agricultural production demands, researchers estimate 90% of that production increase will need to occur on currently cultivated land. In eastern Africa, smallholders are the primary agricultural producers (Schultz et al., 2005; Livingston et al., 2011). In smallholder semi-arid rainfed agriculture, supplemental irrigation can provide resilience in dry spells, therefore improving yields and potentially providing more profit for farmers through production of high-value market crops (Dile et al., 2013; Gower et al., 2016).

Typical smallholder irrigation schemes include pumping from nearby rivers, surface runoff collection into small ponds, shallow boreholes, and rainwater harvesting, with conveyance through open channels, flexible pipes and buckets (Nakawuka et al., 2018). In Kenya, approximately 87% of irrigation is sourced from surface water and 13% is from groundwater (FAO, 2020a). Low-cost drip kits were introduced in Kenya in 1995, and Laikipia County appears to endorse this technology for sustainability (Laikipia, 2020b). However, there are many barriers to improved irrigation, including land tenure issues, lack of electricity and infrastructure, lack of awareness and Agricultural Extension, lack of reliable markets, lack of access to financial and credit services, and overdependency on NGOs (Nakawuka et al., 2018). Specifically, a lack of Agricultural Extension Services means that regionally specific agricultural research knowledge does not reach farmers and ultimately resources that could be conserved are wasted (Emmanuel, 2012; Smith et al., 2014).

To address these barriers to irrigation adoption, Kenya has been moving towards a polycentric water governance system where multiple local, regional, and national authorities interact to make management decisions based on each regions social and ecological conditions (Baldwin et al., 2015). In Laikipia County, the Laikipia Water Conservation Strategy, LWCS (2014-2018) was developed to balance the needs of land users and the ecosystem while highlighting the roles of local and national government, researchers, and other stakeholders for implementation (Laikipia Wildlife Forum, 2013). This LWCS document encourages drip irrigation, water harvesting and storage, dam and borehole rehabilitation and construction, and diversifying income generation outside of irrigated farming to reduce water use and ameliorate water shortages in the area (Laikipia Wildlife Forum, 2014). Ngigi et al., (2008) studied flood storage as an irrigation technology in a Laikipia sub-catchment and found that it had the potential to reduce erosion during extreme events and capture enough water to sustain agriculture and river flows during dry seasons, but the infrastructure investment would be significant.

Ultimately, improved irrigation efficiency and conservation tillage are two different methods of conserving on-farm water, with irrigation water being stored in tanks or ponds for later use and conservation tillage storing water in the form of increased soil moisture (Jaegermeyr et al., 2016). Due to the different mechanisms of storage, the corresponding streamflow response of each practice may vary along the steep rainfall gradient in the Laikipia region, from upstream Mount Kenya to the lower downstream savanna. We pose that these dynamics may follow a similar pattern as those that govern savanna vegetation (Campo-Bescos et al., 2013, 2015; Southworth et al., 2018), as explained in the next section.

1.3 SAI and Savanna Ecosystems

Savanna landscapes are characterized by the existence of trees, grasses, and scrub at varying densities (Scholes and Archer, 1997). While vegetation dynamics and climate are linked through soil moisture, other factors, including mean annual precipitation, rainfall intensity, temperature, fire, potential evapotranspiration, and herbivory dominate the dynamics at different locations along the rainfall gradient (Sankaran, 2005; Good and Caylor, 2011). Campo-Bescos et al. (2013) used dynamic factor analysis (DFA) to assess drivers of vegetation (through relationship to NDVI) spatially along the savanna rainfall gradient. They found soil moisture and precipitation were important factors in low-rainfall (<750 mm) regions; temperature, evapotranspiration, and fire were important factors at high-rainfall (>950 mm) regions; a transition in the importance of soil moisture and precipitation in medium-rainfall (750 mm to 950 mm) regions (Campo-Bescos et al., 2013) (See Figure 1-1).

Although theoretical arguments have been made for SAI protection of ecosystems, few studies exist that quantify potential benefits of converting from conventional to SAI practices and connect those benefits to key, and sometimes spatially remote, ecosystem services or biodiversity in the region of interest. Since Laikipia County considers wildlife tourism in the downstream savannas as an economic pillar, a holistic approach to policy recommendations to ensure that both farmers and remote wildlife benefit is critically needed. In addition, Laikipia County has many private ranches in the downstream savannas that also act as conservation lands, meaning streamflow and quality in this area is equally important for livestock and wildlife.

Different savanna animal indicator species have diverse water requirements. Typically, livestock in northern Kenya need access to water every 1-3 days (Coppock et al., 1988). Plains zebras access water every 1-2 days (Cain et al., 2011), while Grevy’s zebras have slightly more drought tolerance and can wait up to three days for water access (Churcher, 1993). Most species are also able to adapt to short droughts using ephemeral watering holes throughout the landscape, while species such as the waterbuck are highly dependent on surface water availability in streams and rivers (Smith et al., 2006). Therefore, connecting upstream SAI to downstream changes in hydrology, including flow volume, quality, and hydroperiod impacting wildlife and livestock is critical to begin quantifying ecosystem benefits of SAI adoption.

1.4 Hypotheses and Objectives

Although both conservation tillage and improved irrigation efficiency may improve streamflow and quality locally when compared to conventional agriculture, the literature presented above could also indicate that the significant impact of agriculture in the area is too great to sustain wildlife and livestock located in the remote downstream savanna in dry seasons and drought years.

Grounded on the previous literature, the overarching research question for this dissertation is: *How do SAI practices (conservation tillage and increased water use efficiency) impact ecosystems at the watershed scale?*

1.4.1 Hypotheses

This dissertation aims to answer this overarching research question through two relevant and testable hypotheses, *H1* (with sub-hypothesis *H1.1)* and *H2*:

H1. Across the savanna landscape gradient from high (>950 mm annually) to low (<700 mm annually) rainfall, general conversion from conventional to conservation tillage in intermediate rainfall areas has a larger relative positive impact on streamflow magnitude and duration than conversion in high rainfall areas..

> *H1.1* (sub-hypothesis). Considering equivalent areas of adoption of two alternative SAI practices, conservation tillage and efficient irrigation a combination of

efficient irrigation adoption in high rainfall areas with conservation tillage adoption in intermediate rainfall areas will minimize impacts on watershed-scale streamflow magnitude and duration compared to any other spatial combination of practice adoption.

H2: Beyond the positive impact (mitigation) on the agricultural region, the conversion to improved SAI practices on the same agricultural footprint will still not be sufficient to sustain downstream savanna ecology for livestock and wildlife. The ecosystem impact will be related to changes in the hydroperiod and flow volume.

1.4.2 Specific Objectives

The following objectives were developed to test hypothesis H1/H1.1:

1. Develop the new CUENCA model and evaluate its efficiency to simulate current streamflow and quality conditions using available data. This is needed to create a sufficiently sensitive and parsimonious model for evaluation of SAI impacts.
2. Identify watershed/regional physical responses to conservation tillage and improved irrigation efficiency in similar soil and climatic conditions and verify that CUENCA is sufficiently sensitive to reflect potential management changes (i.e. scenarios) via streamflow indicators in the Mount Kenya region. This will serve to demonstrate if CUENCA can capture t biophysical changes as a result of SAI adoption.
3. Develop spatially distributed agricultural management scenarios across the rainfall gradient (low, medium, and high rainfall) to depict farm conversions to conservation tillage and improved irrigation efficiency at different adoption levels, regardless of farm size and crop, and translate these scenarios into CUENCA input parameters. This will provide a basis for evaluation of SAI at different adoption levels and along the rainfall gradient and at varying landscape positions.

4. Quantify and analyze differences in streamflow, including volume, peak flow, flow duration, hydroperiod, and quality, among scenarios at key locations in the watershed during dry and rainy seasons; and dry, wet, and average years. This is necessary to identify measurable and statistical differences among scenarios and ultimately evaluate effects of SAI.
5. Identify tipping point or crossing point along the rainfall gradient at varying levels of adoption where the two practices have an equivalent positive impact on the river. This will inform land managers, NGOs, and other stakeholders where along the rainfall gradient to concentrate efforts of SAI.

The objectives for testing sub-hypothesis 2H2 include:

1. Identify key species of wildlife and livestock present in the downstream savanna area that have different sensitivities to water availability (i.e. duration of drought) and water quality and different mobility ranges. This is necessary to assess whether a hydrologic change translates to direct impacts on ecosystems.
2. Identify potential metrics that can have impact on wildlife and livestock performance (number of consecutive days below stream flow threshold; frequency of days below threshold) that can be modeled using CUENCA. This will provide important baselines of comparison among scenarios.
3. Link key species to changes in ecosystem function as a result of SAI practices and assess how adoption of SAI practices could potentially alter livestock and wildlife performance (mobility and survival). This will provide a basis for land managers, NGOs, and other stakeholders to develop realistic sustainability and ecosystem protection goals.

1.5 Dissertation Sections

This dissertation is composed of 5 chapters. This first chapter introduces the general background of sustainable agricultural intensification and hydrologic modeling of such systems, as well as the hypotheses and objectives of this research. The second chapter describes hydrological model development and testing for sensitivity to tillage and irrigation. The third chapter first details a complete field and remote sensing dataset for Laikipia, Kenya gathered in this research that can be used to test hypotheses about agricultural management impacts on biophysical variables across a heterogeneous watershed. It then field tests and applies the CUENCA model to evaluates SAI management adoption scenarios in Laikipia, Kenya and analyzes their impacts on drought (river drying) frequency and duration. The fourth chapter evaluates linkages on downstream savanna ecosystem services from SAI adoption scenarios, and specifically the endangered Grevy’s zebra. The fifth chapter summarizes the main conclusions of this dissertation.

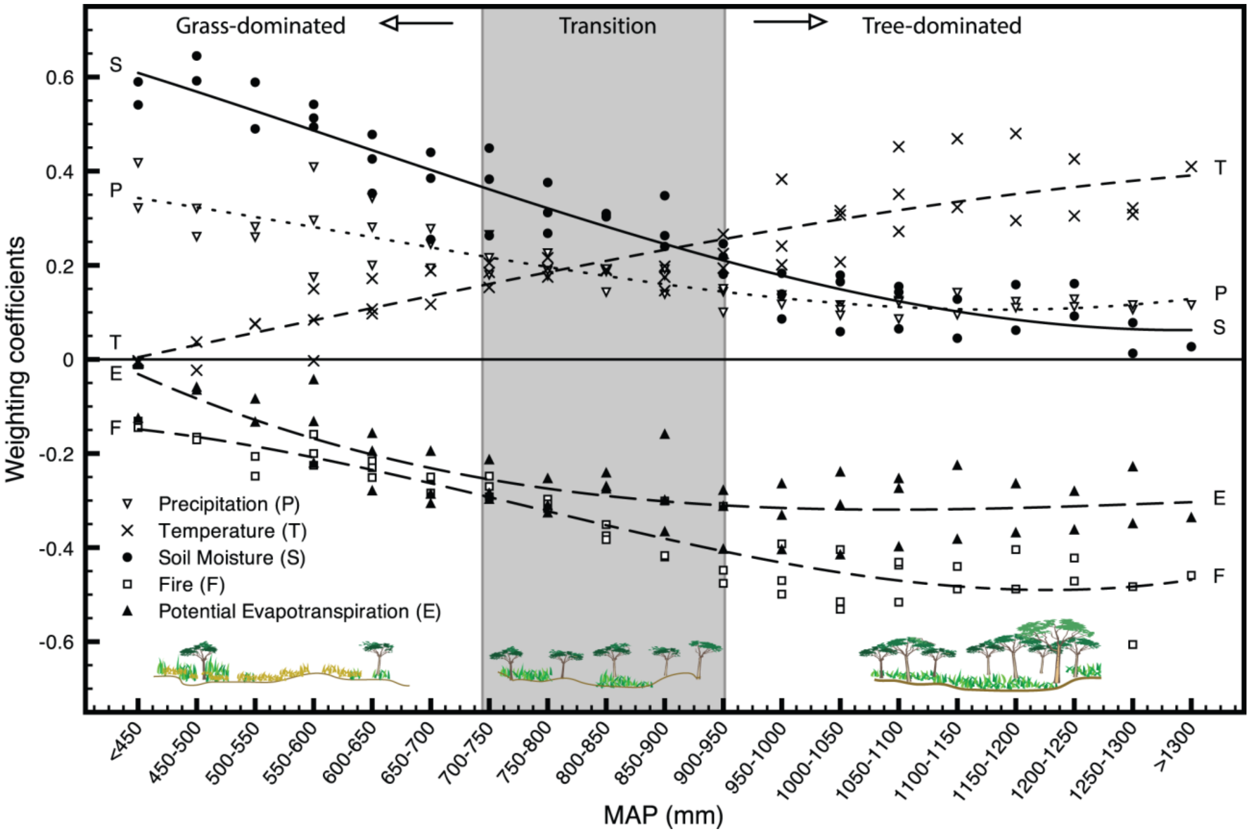

Figure 1-1. Results from Campo-Bescos et al. (2013) detailing drivers of savanna vegetation based on mean annual precipitation (MAP).

CHAPTER 2

DEVELOPMENT OF A PARSIMONIOUS LINK AND NODE HYDROLOGIC MODEL FOR EVALUATION OF SUSTAINABLE AGRICULTURAL INTENSIFICATION PRACTICES

Use of hydrologic models to characterize flow processes and support water management decisions has become standard practice over the last 80 years (Freeze and Harlan, 1969; Farmer and Vogel, 2016). These models can serve a wide variety of purposes, including landuse and infrastructure planning, agricultural water management, and ecosystem service decision support. Hydrologic modelling is complex, attempting to simulate hydrologic processes across different spatial and temporal scales while accounting for landscape and climate variability (Freeze and Harlan, 1969; Bloschl and Sivapalan, 1995; Clark et al., 2016). Over time, frameworks for model development, selection and use were created based on project objectives and hypotheses, data availability, and scale (Bergstrom, 1991). However, many issues of spatial and temporal scaling, model parsimony, and model equifinality persist. (Clark et al., 2017; Beven, 2006

Process-based models are based on physical earth processes derived from first principles and data collected in the field (Freeze and Harlan, 1969; Clark et al., 2011). In the United States, the United States Geological Survey (USGS), National Oceanic and Atmospheric Administration (NOAA), and United States Department of Agriculture Natural Resources Conservation Service (USDA-NRCS) have compiled long-term and relatively high-resolution streamflow, weather, and soils datasets that can be coupled with empirical and/or mechanistic models to estimate evapotranspiration, infiltration, and groundwater recharge to develop watershed models (HCDN, 1992; Peschel et al., 2006; Brakenridge and Anderson, 2006). As high spatio-temporal resolution satellite products from missions such as Landsat and Sentinel have become readily available, as well as a re-analysis products based on satellite imagery and corrected using ground observations, field observations, which can be time consuming and expensive to obtain, have become less common in hydrology (Barthold and Woods, 2015; Burt and McDonnell, 2015). In addition, data-driven and machine learning models are becoming common tools for streamflow forecasting (Frame et al., 2023; Muñoz-Carpena et al, 2023). However, to evaluate impacts of landuse or management change under a changing climate, these models may lack important dynamics linked to physical processes such as subsurface and riparian storage, vegetation dynamics, and extreme weather events (Birkel et al., 2011).

Concurrently, common models used by water managers have become overly parameterized and also typically underperform during extreme events or when extrapolated to watersheds or scenarios for which no calibration data was available (Kirchner, 2006). This leads to problems of model equifinality, when multiple sets of inputs lead to the same model outputs because there are so many parameters and input uncertainty bounds (Beven, 2006). While a truly parsimonious model (i.e. “toy models”) may contain very few parameters, to represent complex filed dynamics it often necessary to consider larger sets of parameters. Still, there is value in simple (although not truly parsimonious) process-based models that are reliant on as few parameters as possible while still capturing the complexity of a watershed. Muller et al. (2011) present an interesting discussion on the this tradeoff between model complexity, uncertainty and its sensitivity to respond to specific objectives.

During a modelling exercise, model selection should be based on the project hypotheses and objectives, model limitations and necessary project outcomes. This includes ensuring the model is sufficiently sensitive to the parameters of interest, important site-specific watershed processes are included, the correct spatio-temporal scales are addressed, and that data are available for model calibration and validation (Muñoz-Carpena et al., 2006; Engel et al., 2007). Hydrologic models are categorized as lumped, distributed, and semi-distributed models based on spatial explicitness (Haan et al., 1982)). Lumped models typically average physical values across a watershed and are linked by stream channels. Distributed models use gridded input datasets to reflect more complex landscape heterogeneity. Semi-distributed models land in the middle, where gridded datasets can be used to characterize a watershed or generate hydrologic responses in sub-basins, and the results are lumped to route water flows and their constituents through the landscape. While distributed models may have an advantage in situations where a problem requires high-level process understanding, they typically suffer from the fact that it is almost impossible to fully identify spatial variability in heterogenous catchments and the subsequently large number of parameters used to capture variability make model calibration, validation, and evaluation extremely difficult (Beven, 1989; Bloschl and Sivapalan, 1995).

Multiple spatial and temporal scale issues are present in hydrologic models that contribute to their overall uncertainty (Shirmohammadi et al., 2006). First, hydrologic processes occur at many different physical scales (e.g. flooding vs soil-water interactions) and temporal scales (e.g. rainfall and evapotranspiration processes). For researchers and resources managers, there is also a limit on the observation scale of these processes, or how often measurements can be taken to characterize current conditions or observe processes. Some behaviors may have daily, seasonal, annual, or multi-year trends that can be missed if there is a mismatch between observation and process scales (Blosch and Sivapalen, 1995). Many hydrologic models are evaluated on monthly timescales due to data requirements (Sudheer et al., 2007), but changes in streamflow can affect ecosystems on a daily and weekly timescale (Vigerstol and Aukema, 2011).

Once an objective is identified and a reasonable scale determined, the model selected should be sufficiently sensitive to the parameters of interest (Bergstrom, 1991) . In sustainable agricultural intensification (SAI), tillage and irrigation practices are frequently recommended to improve management of water resources. Reduced tillage is purported to improve soil structure and increase water holding capacity, while improved irrigation (localized like drip, tapes, microsprinklers, etc.) is assumed to apply water more precisely and efficiently, reducing unnecessary losses to evapotranspiration or increased soil moisture (Van der Kooj et al., 2013). The Soil and Water Assessment Tool (SWAT) and Annualized Agricultural Non-Point Source (AnnAGPS) are the most common models used to assess potential benefits of conservation practicesand were employed during the USDA-ARS Conservation Effects Assessment Project devised to quantify the effects of conservation practices across USA working lands (Tomer and Locke, 2011). Although globally used, SWAT may be better suited to predicting streamflow in humid climates (Veith et al., 2010). In addition, SWAT is more sensitive to parameters (curve number, crop rotations, and soil coverage) other than those representing tillage practices, meaning that the impacts of changes in tillage may be underestimated when compared to scenarios of landuse change, crop rotations, or soil coverage (Ullrich and Volk, 2009). Bowmer (2011) discusses the difficulty of attributing changes in river flow and quality in a watershed to specific agricultural practices or land use changes, with a specific discussion on scale and lag-time. SWAT tends to perform better at a monthly timescale (Sudheer et al., 2007).

The objective of this work is to develop a parsimonious watershed model sensitive to AISAI spatial adoption changes (i.e. irrigation nd tillage). Thus, CUENCA link and node model was developed to address concerns of temporal scale and tillage sensitivity in agro-ecosystems. CUENCA consists consist of a link-node system based on physical processes (Hromadka et al. 1985, Muñoz-Carpena & Parsons, 2004) where the watershed is divided into unique hydrological and land use sub-basins contributing water to nodal points that are linked by different hydrological processes: rainfall-runoff to the sub-basin outlet is based on rural land-use type (USDA−NRCS, 1986), channel flow and stream routing between pairs of downstream nodes, flow-by structures (i.e., river/canal plus lateral seepage or extraction into agricultural ponds), flow-through structures (reservoir), pipes, and water use abstractions (crop water and rural water use, and stream seepage). The sub-basin characteristics (farm and other land use, crop rotations, management practices, topography, climate, soils) are obtained from existing remote sensing product and local field measurements as shown in the next Chapter for the Laikipia River watershed, Kenya.

2.1 Methods

Figure 2-1 outlines the methods used in this chapter. First, the CUENCA link and node model was developed for agricultural and urban hydrology simulations. Then, the model sensitivity to different rainfall-runoff algorithms, curve number and Green-Ampt was tested. Next, sensitivity to tillage and irrigation was verified.

2.1.1 Model Development and Processes

The conceptual basis of CUENCA was adapted from Hromadka et al. (1985) for an event-based link and node model utilizing rational and obsolete ss-curve methods to model runoff in urbanized areas. CUENCA is written in modern Fortran as an structured code for continuous simulation and extended functions and processes for agro-ecological watersheds, while maintaining low data requirements compared to other common process-based models. This provides a tractable simulation tool to analyze the observed streamflow dynamics, with a flexible link and node approach where the user can assign a variety of common hydrological and sediment transport process. Figure 2-1 includes a chart outlining the methods for this entire chapter. Figure 2-2 includes a flowchart of the model, including inputs, process linkages, and outputs, and Figure 2-3 details the baseflow and percolation processes.

Rainfall-runoff processes. To simulate rainfall-runoff processes, two algorithms were implemented in CUENCA whereby users can choose between the Curve Number (CN) (USDA-SCS, 1985) and Green-Ampt (Green and Ampt, 1911) (GA) methods. Daily rainfall input values are disaggregated within the model into 5-min rainfall hyetographs using the SCS alternating block methods based on standard cumulative storm types (I, IA, II, III).

The CN method calculates runoff volume per event based on the SCS curve number, which is a function of watershed landcover, management, antecedent moisture condition, and soil type (specifically hydrologic soil group). If a watershed has multiple landcover or management types, an area-based weighted curve number can be used. The method includes an adjustment for CN based on the previous 5-days' precipitation and irrigation amounts (i.e. adjusted to CN-I when previous 5-days' effective precipitation is less than 36 mm and adjusted to CN-III when greater than 53 mm), also called the antecedent moisture condition (AMC). Time of concentration is calculated based on curve number, flow path length, and watershed slope. Once runoff volume is calculated using the CN method, the remainder of the rainfall-runoff process follows the method outlined in the TR-55 manual: peak flow is calculated and used to scale the SCS unit hydrograph; the unit hydrograph and excess hyetograph are used to develop a convolution hydrograph of the runoff at a 5-minute timestep (Chow, 1987). The base code for this component was imported from the model UH (Muñoz-Carpena and Parsons, 2004).

The Green-Ampt (GA) component runs at a 5-minute timestep to generate runoff based on unsteady rainfall and alternating between ponded and non-ponded conditions (Mein and Larson, 1973; Chu, 1978; Skaggs and Khaheel, 1982). Important inputs to the GA component include saturated hydraulic conductivity (Ksat), suction at the wetting front (Sav), saturated water content (Theta S), initial water content (Theta Ini), wilting point (WP), and field capacity

(FP). Time of concentration is typically sensitive to drainage area, slope, basin shape, length of flowpath, and Manning's N. To address all sizes as well as urban and rural watersheds, the user can choose among five different methods to calculate time of concentration (Williams (Williams, 1922), Johnstone-Cross (Johnstone and Cross, 1949), Bransby-Williams (Abustan et al., 2008) Passini (Salimi et al., 2016), or the ensemble of values as a default, which removes the minimum value and averages the remaining three valued). Peak flow is determined using the NRCS triangular unit hydrograph. Finally, the daily direct runoff hydrograph is obtained by convolution of the 5-minute timestep triangular pulses. The base code for this component was ported from the model VFSMOD (Muñoz-Carpena et al., 1999).

Irrigation. The irrigation process is tied to the precipitation process in the CUENCA model. Users can input information about irrigation depth, frequency, and percent of coverage over the watershed for each subwatershed. The irrigation depth is multiplied by the percent coverage so that irrigation is not spatially explicit at each subwatershed, but rather lumped and averaged over the whole watershed as the rest of the landscape parameters are. Users can specify when to simulate irrigation based on previous days' rainfall amounts plus the current day's rainfall. So, a user can specify to irrigate if the previous rainfall 3 days plus the current day's rain do not exceed 25.4 mm. If the simulation inputs include conventional irrigation depths of 25.4 mm, this means irrigation would occur every 4 days if there is no rainfall. If the drip irrigation scenarios are being evaluated (e.g. irrigation depths of 6.35 mm), then irrigation would occur daily when no rainfall occurs. Irrigation depth is added to effective precipitation and therefore each irrigation event is treated as a storm event in the CUENCA model and can change the AMC condition for CN calculations.

Evapotranspiration, baseflow, and groundwater recharge. Both rainfall-runoff processes are linked to a rootzone infiltration model, ThetaFAO (Munoz-Carpena, 2012), which partitions shallow root-zone infiltration into evapotranspiration, crop water, and percolation based on FAO-56 detailed estimation of adjusted evapotranspiration under soil water stress limiting conditions (FAO, 1998). The daily soil percolation (below root zone) volume is further divided into baseflow and recharge based ACRU agro-hydrologic model (Schulze, 1995). In both soil storage compartments (Fig. 2-3), water volume that exceeds field capacity exits immediately as baseflow, and similarly no water is lost if water content drops below wilting point. Evapotranspiration does not occur on days with rainfall, however in this future this may need to be considered in the context of storm duration rather than simply a rainfall event.

Evapotranspiration has the potential to be a significant process in a watershed water balance. In irrigation studies, it is one of the primary factors for water use efficiency and irrigation flow partitioning (Grafton et al., 2018; Scott et al., 2013). In this model, since it directly affects shallow root zone water content, it also affects storm-event runoff volume.

Channel flow. Streamflow routing (convex) – A form of the convex channel routing method (SCS, 1972) is used to develop hydrographs for open channel flows in CUENCA. This is based on a channel routing coefficient, C, which can be estimated as a function of the average flow velocity at the upstream portion of a channel, or the velocity when the channel is flowing at normal depth. At each node, an outflow hydrograph is estimated from an inflow hydrograph using channel parameters including slope, flow path length, average velocity, and Manning’s n. This method is best suited for urban hydrology where in-stream losses and long inconsistent reaches are minimal, rather than natural channel systems. (Hromadka et al., 1985). To extend to larger areas, aa linear in-stream loss function was added to CUENCA, where the user has the option to set the loss function or lump losses in with baseflow depending on the hydrology of the system. The convex process also contains a streamflow hydrograph recession process based on. The two recession coefficients can be derived from local stream flow data when available or use general numbers (Brutsaert and Nieber (1977). The increase or decrease in baseflow between two stream nodes is then used to re-calculate soil moisture in the contributing watershed as initial moisture condition for the rainfall-runoff and mass balance calculations.

A pipeflow process is available for pipe diversions or inflows on the main stream channel. It is similar to the convex stream-routing process where an outflow hydrograph is developed from an inflow hydrograph based on pipe length, diameter, slope, and Manning's n. If the pipe is flowing at full-capacity, all excess water is retained behind the pipe, which is adequate for most storms but may not accurately represent flooding events where flows exceed road or bridge elevations. In both pipeflow and convex processes, CUENCA does not consider backwater effects as a simplification (Hromadka et al., 1985).

Four additional node processes are included (Fig. 2-2) to allow the user to match the topology of the local hydrological network. The "add" process simulates a confluence of stream channels or pipes by adding hydrographs at a node. The "split" process fractionally splits streamflow into two different streams at a node. The flow that is removed from the primary stream can be added to another stream specified by the user within this same process. The "move" process moves flows forward in time by a specified increment. This process is only used if a channel is assumed to attenuate no flow (i.e. peak flow rate remains the same) and a simple calculation of shift in time to peak flow can be used as a routing model. Finally, the "clear" process removes all flow and data from a stream and serves to reinitialize the system as needed.

Detention and retention basins. A detention basin in the watershed can be simulated using the "flow-through" process that uses the Modified-Puls Method (Chow, 1964; Henderson, 1966). In this process, only water that fills the dead storage is permanently stored, while the rest is attenuated in the basin before flowing back to the same stream. The user can input up to 5 basin data points that relate basin depth to storage and outflow. In addition, the user can specify total dead storage, and dead storage and effective volume at the beginning of a simulation.

Another process can be used to simulate wetlands or large recreational ponds providing flood protection in watersheds. The "flowby" process stores flows exceeding a maximum threshold in a retention structure or sends them to another stream channel. The user selects the maximum flow velocity and whether or not the stream is stored in a permanent reservoir or another channel used within the model. Values to parametrize "flow-through" or "flowby" features can be obtained readily from hydrographic maps.

2.1.2 Description of Inputs

Each node has a suite of inputs specific to hydrology, weather, and management in the watershed draining to the node (the link area). These data are fed to the model through discrete input files organized by node. The hydrology files include baseflow and existing streamflow (i.e. spring flow or snowmelt). For the weather data, CUENCA requires a set of daily inputs for each subwatershed corresponding to each node. If a gridded dataset is used, then the inputs will be lumped by subwatershed to create a semi-distributed model. Figure 2-3 shows a watershed and its corresponding sample link and node system. Required daily inputs are: precipitation (mm), reference evapotranspiration (ETo) (mm), minimum temperature (degrees C), maximum temperature (degrees C), average wind speed, mid season crop coefficient (CKmid), initial soil moisture (m^3/m^3), and daily average snowmelt or springflow (m^3/s). If it is known, management information such as irrigation (mm) and water abstraction (mm) can be included as well.

In addition to these input files, the input file that calls the processes and links the nodes throughout the watershed requires information specific to each node and process as well. Table 2-1 includes the data required for each process. Appendix D includes sample input and output files, and Appendix E contains the CUENCA Fortran code.

2.1.3 Description of Outputs

CUENCA outputs are aggregated at a daily timescale, and include the following values at each node (m3): precipitation, incoming streamflow, outgoing streamflow, runoff, baseflow, actual evapotranspiration, groundwater recharge, permanent losses (to other watersheds, wetlands, or retention ponds), total infiltration, changes in root zone soil water content, and changes in percolation soil water content.

2.1.4 Evaluation of Model Sensitivity

The CUENCA model was evaluated for its sensitivity to tillage and irrigation practices, and the differences in these sensitivities when either the GA or CN method is used to simulate rainfall-runoff processes. Figure 2-1 outlines all sensitivity tests described in this chapter. This is a critical model feature desired for testing SAI adoption in the current study. For all scenarios, watershed areas of 100 ha with slopes of 3% and longest watercourse of 2000 m were used in simulations. Depending on the test, a variable dataset of 30 days of rainfall events (all less than 30 mm) or events of 6.35, 12.7, 25.4, 50.8, 76.2, and 101.6 mm were used.

Event-based sensitivity to runoff method. First, a simple single-event comparison of runoff volumes using GA and CN was conducted. Green-Ampt parameters were selected from Rawls and Brakensiek (1983) and Saxton et al. (2006) to simulate conditions under clay, sandy clay loam, and sandy loam soil textures and bare surface conditions. Once runoff depth was determined using a GA scenario for each soil texture, a NRCS curve number value was selected to match the runoff depth (mm) for the storm. The input values specific to each soil texture are shown below in Table 2-2. Simulations were run using 25.4 mm of rainfall for clay and sandy clay loam, and 100 mm of rainfall for sandy loam.

Then, a 30-day time series simulation was performed for clay soil using varying precipitation depths all under 30 mm. The same initial values and input parameters were used for GA and CN methods, and the time series values were plotted to evaluate daily streamflow for each simulation. A graph was also made with event-based partitioning into infiltration and runoff throughout the time series. Clay was selected for testing in anticipation of model application in Laikipia, Kenya, which contains primarily clay soils.

Evaluation of sensitivity to initial soil moisture. Next, the CN and GA methods were evaluated for sensitivity to initial soil moisture conditions. Once again, clay was selected because of near-future model applications. These simulations used initial moisture conditions of 0.42 m^3/m^3 (field capacity), 0.36 m^3/m^3 (halfway between field capacity and wilting point), and 0.30 m^3/m^3 (wilting point). Varying rainfall depths of less than 30 mm were used as the time series input.

Evaluation of irrigation sensitivity. Clay soils were tested for sensitivity to irrigation under the CN and GA algorithms. Initial values for conventional tillage in Table 2-4 were used to characterize the soils, and rainfall was equal to zero during the scenario, which was run for 30 days. Because the CUENCA model uses upstream flow as the source for irrigation, constant baseflow of 2 m^3/s was used as an input to the model. Irrigation scenarios of 6.35 mm, 12.7 mm, 25.4 mm, 50.8 mm, 76.2 mm, and 101.6 mm were used to simulate a range from drip irrigation (6.35 mm) to standard irrigation (25.4 mm) to possibly deep furrow irrigation (101.6 mm). The change in maximum streamflow and minimum streamflow over the study period was plotted against irrigation depth, as well as the normalized change in streamflow vs normalized irrigation

Evaluation of tillage sensitivity. Different tillage regimes were evaluated under clay soil types only. First, the curve number was adjusted according to typical decreases based on no-till or reduced tillage (Sur et al., under review) and corresponding residue retention associated with each regime (Rawls et al., 1980). Therefore, the curve number was reduced by 9% from conventional tillage (CT) to no-till (NT), and 7% for reduced tillage (RT). Green-Ampt parameters under reduced and no-till were determined by finding parameters that fit an equivalent runoff from the 25.4 mm (1 inch) storm for the tillage-adjusted curve numbers. Then, sensitivity to tillage under different storm events was tested by running a single event at precipitation depths of 6.35 mm, 12.7 mm, 25.4 mm, 50.8 mm, 76.2 mm, and 101.6 mm. Rainfall was then plotted against runoff and the runoff coefficient (runoff/precipitation) for each scenario.

2.2 Results

Event-based sensitivity to runoff method. The results of a single event simulation are shown in Figure 2-5. Runoff events generated using the GA process tend to have higher peak flows and shorter runoff durations. This can most likely be attributed to the explicit accounting used to generate a runoff hydrograph in the GA method, which takes into account instantaneous infiltration over five-minute timesteps, as well as a different time of concentration (Tc) calculation method as described in the methods. This difference in Tc biases peak flow timing between the two methods. Table 2-3 contains the difference in runoff volumes generated under the different processes, which are all less than a 0.23% difference, with sandy loam having the smallest difference of 0.02%.

Figure 2-6 (top) shows the difference in daily runoff volume in clay and sandy clay loam soils over 30 days of variable rainfall. While very little runoff is generated in the sandy clay loam scenario, the clay scenario indicates that the GA method generates significantly more runoff during a rain event than the CN method over time. This is most likely due to the continuous GA physical response to initial soil moisture content, whereas CN partitions rainfall into runoff or infiltration empirically based on discrete antecedent moisture condition (I, II, III), which serves as a proxy for soil moisture. Table 2-4 shows the breakdown of water balance components over the 30-day simulation, and CN has higher rates of baseflow and soil moisture losses (i.e. baseflow losses, in this instance). Due to the functioning of the baseflow process in CUENCA, which in this case is initialized when soil field capacity is exceeded, this baseflow is all occurring during storm events. Therefore, streamflow response would show higher peaks for the CN process that GA process during large storm events. Over the 30 day period, the total water balance differs by 0.8%. Figure 2-6 (bottom) shows the depth of runoff or infiltration during each storm in the clay soil simulation. Other than the largest rainfall events, almost all rainfall in the CN simulation infiltrates into the soil. While CUENCA does contain a soil moisture process so that water more than field capacity flows to the stream as baseflow, the CN process may artificially increase soil moisture compared to the GA method because GA responds to both instantaneous infiltration rates and soil moisture during rainfall partitioning. Therefore, in a heavy clay soil with a low infiltration rate, under intense rainstorms, we expect GA may more accurately represent rainfall-runoff response if the effective soil hydraulic properties are known. The full water balance for these simulations is contained in Appendix A.

Evaluation of sensitivity to initial soil moisture. Figure 2-7 shows a time series of direct runoff volume under different initial moisture conditions for a clay soil. These simulations used initial moisture conditions of 0.42 m^3/m^3 (field capacity), 0.36 m^3/m^3 (halfway between field capacity and wilting point), and 0.30 m^3/m^3 (wilting point). While CN runoff volumes only varied on the first day due to CN alterations for different antecedent moisture conditions, GA runoff volumes vary throughout the time series.

Evaluation of irrigation sensitivity. Figures 2-8 and 2-9 show the results of the evaluation of the sensitivity to irrigation in the CN and GA processes. Figure 2-8 shows the total change in streamflow as well as normalized values. Typically, as irrigation increases, streamflow decreases as expected. The only exception to this is when the Curve Number method is used for irrigation greater than 53 mm, because this is when a curve number adjustment is made based high antecedent moisture condition. Figure 2-9 shows a time series over 30 days of no rainfall. At low irrigation levels (i.e. 6.35 mm), which might be typical of a drip irrigation scheme, streamflow stays continually at a decreased level. At higher irrigation levels, when fields would typically be irrigation less frequently, there are reductions in streamflow followed by a return to normal. In this simulation, constant baseflow from upstream is provided as a source for irrigation, otherwise a return to typical flow levels may take longer.

Evaluation of tillage sensitivity. Figure 2-10 shows the results of varying tillage regimes. Table 2-5 shows the inputs used in the scenarios for GA and CN. During smaller rain events (<25.4 mm), the Green-Ampt process produces more runoff and has a higher runoff coefficient under each tillage regime than CN. As rainfall increases, the CN runoff coefficient increases. CN runoff volume for NT scenarios is greater than even the RT scenarios under GA. Figure 2-11 shows time series of runoff under varying rainfall events. In this example, Green-Ampt typically has higher runoff volumes, and all of the precipitation events simulated were less than 25.4 mm.

2.3 Discussion

Previous comparisons of CN and GA in SWAT at a daily timescale contained similar results, with GA having higher peak flows (although not as different as those observed in this work) and higher cumulative streamflow over time (Ficklin and Zhang, 2013).

The results of the model testing indicate that both the CN and GA methods perform similarly, but GA is more sensitive to initial soil moisture. This means that runoff will occur at times after a small rain event, which is important to when modeling clay soils. Since most of the simulated storms in these tests were less than 25.4 mm, further evaluation should be compiled for larger storm events. The current version of SWAT updates CN based on soil moisture, rather than the AMC method used in this manuscript (Kannan et al., 2007). Therefore, while SWAT was traditionally highly sensitive to CN, some studies now indicate that it is watershed specific (Lenhart et al., 2002; Kannan et al., 2007).

The methods also have slightly different results for irrigation application, and as application rates increase, subsequent streamflow is less impacted using the GA method. This should be tested with different soil textures. Overall, more rigorous sensitivity testing should be performed to evaluate the model and other processes within the model.

Results also indicate that CUENCA is sensitive to different tillage scenarios. Ullrich and Volk (2009) found that SWAT was less sensitive to tillage intensity when compared to sensitivity of duration of vegetation soil coverage, timing of planting and first tillage only, and conservation support practices like contouring. However, CUENCA is simple compared to SWAT and these inputs are not available to change at a daily time scale in CUENCA.

In all, CUENCA exhibits the required sensitivity to SAI practices (tillage and irrigation) needed to analyze impacts of the spatially explicit adoption of these practices in the watershed.

2.4 Conclusions

The CUENCA link and node model was developed to connect land and water management practices to far reaching ecosystem services at a daily timescale. The option for the user to choose flexible node locations as well as the Curve Number or Green-Ampt rainfall-runoff process and a wide range of hydrological processes gives the user some control over how much data is required to run the model.

The Curve Number and Green-Ampt processes present in CUENCA respond differently to SAI scenarios. On an event basis, the Green-Ampt process tends to display higher peaks in runoff and deplete more streamflow when using irrigation water. However, because GA is physically dependent on soil moisture as an initial condition, it also has more dynamic behavior when used on a multi-day simulation.

These simulations indicate sensitivity to tillage and irrigation. The model will be tested on an existing watershed and field data on the next Chapter. Future studies should aim to test the model under different conditions of SAI practices for validation. There are limits on all model projects, and determining the optimal level of complexity without introducing too much uncertainty is important for simulating ecosystems in this context. After extensive global sensitivity and uncertainty analysis, certain aspects of CUENCA could be simplified or removed. On the other hand, once some case-studies are completed, we may need to add complexity if an important process is missing.

The objective for the CUENCA model to simulate daily streamflow for ecosystem assessment function using relatively modest data requirements compared to other physical models is a difficult one to achieve. Compared to common models like SWAT used for agricultural watershed analysis, CUENCA has low data requirements (Vigerstol and Aukema, 2011).

Table 2-1. Inputs required for each process in CUENCA hydrologic model.

Process	Inputs	Units
Runoff (CN)	Precipitation amount and duration	mm
	Surface storage coverage (swamps, ponds)	fractional
	Landcover/landuse (used in CN calc)	%
	Hydrologic Soil Group (A,B,C,D) (used in CN calc)	%
	Agricultural practices - Planting seasons, crop rotations, field layout (i.e. row crops vs terracing) (used in CN calc)	%
	Elevation, average slope	m, m/m
Runoff (GA)	Precipitation amount and duration	mm, hours
	Vertical saturated hydraulic conductivity	cm/hr
	Average suction at wetting front	cm
	Saturated water content	m3/m3
	Initial water content	m3/m3
	Maximum surface storage	cm
Flowby	Maximum Flowby Q	m3/s
Flowthrough	Dead Storage volume	m3
	Initial dead storage volume	m3
	Initial basin volume	m3
Piper	Length of Pipe routing	m
	Manning's n	unitless
	Change in elevation	m/m
	Pipe diameter	m
Convex	Channel Routing coefficient	unitless
	Channel average flow velocity	m/s
	Base width	m
	Vertical sideslopes	Horizontal:vertical
	Channel length	m
	Manning's n	unitless
Split	Percentage of stream to be diverted	%
Root Zone Infiltration/ Shallow Infiltration/ Groundwater Recharge	Climate (temperature, wind speed)	C, cm/s
	Top soil field capacity water content	m3/m3
	Top soil wilting point water content	m3/m3
	Vegetation properties (Rooting depth, extractable water, height)	m, fractional, m
	Subsoil porosity	m3/m3
	Elevation of streambed in subwatershed	m
	Soil texture	--
Sediment	Soil Texture	--
	Soil Erodibility	Kg/N * h/m2
	C Factor	unitless
	P Factor	unitless
	Sediment size, d50	cm

Table 2-2. Equivalent Green-Ampt and Curve Number parameters used in irrigation evaluation.

	Green-Ampt only		Green-Ampt and Curve Number			Curve Number only
Soil Type	Ks (m/s) 10-6	Sav (cm)	Theta S (m3/m3)	Field Capacity (m3/m3)	Wilting Point (m3/m3)	CN
Clay	0.463	31.63	0.475	0.42	0.30	82.6
Sandy Clay Loam	0.833	21.85	0.398	0.27	0.17	43.2
Sandy Loam	6.06	11.01	0.453	0.18	0.08	20.6

Table 2-3. Event-based percent differences in CN and GA processes.

Soil texture	CN (m3)	GA (m3)	% Difference
Clay	240872	241355	0.20
Sandy Clay Loam (SaCL)	56408	56534	0.22
Sandy Loam (SaL)	280265	280310	0.02

Table 2-4. Water balance comparison between CN and GA processes after 30 day simulation (all units are m3) for clay soils.

Method	Precipi-tation	Runoff	Evapo-transpir-ation	BaseF (TF)	Infil-tration	Root zone change in soil moisture	Subsurface change in soil moisture
CN	196800	17475.96	68986	175920	179462	25940	-91382
GA	196800	41042	68986	120900	165309	25940	-50534
Difference	0	-23566	0	55020	14153	0	-40848

Table 2-5. Equivalent Green-Ampt and Curve Number parameters used in clay soil tillage evaluation. Values are based on 25.4 mm (1-inch) equivalent runoff volumes in clay soils.

Scenario	CN	Ksat (cm/hr)	Sav (cm)
Conventional Tillage	74.4	0.060	31.63
Reduced Tillage	69.2	0.094	44.20
No-till	67.7	0.106	48.20

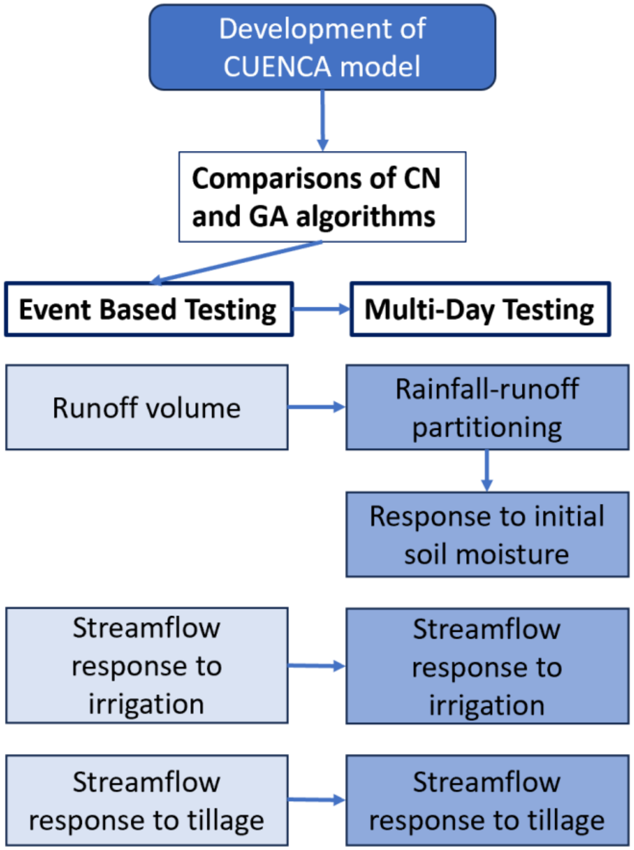

Figure 2-1. Methods for event and multi-day sensitivity scenarios under Curve Number and Green-Ampt algorithms.

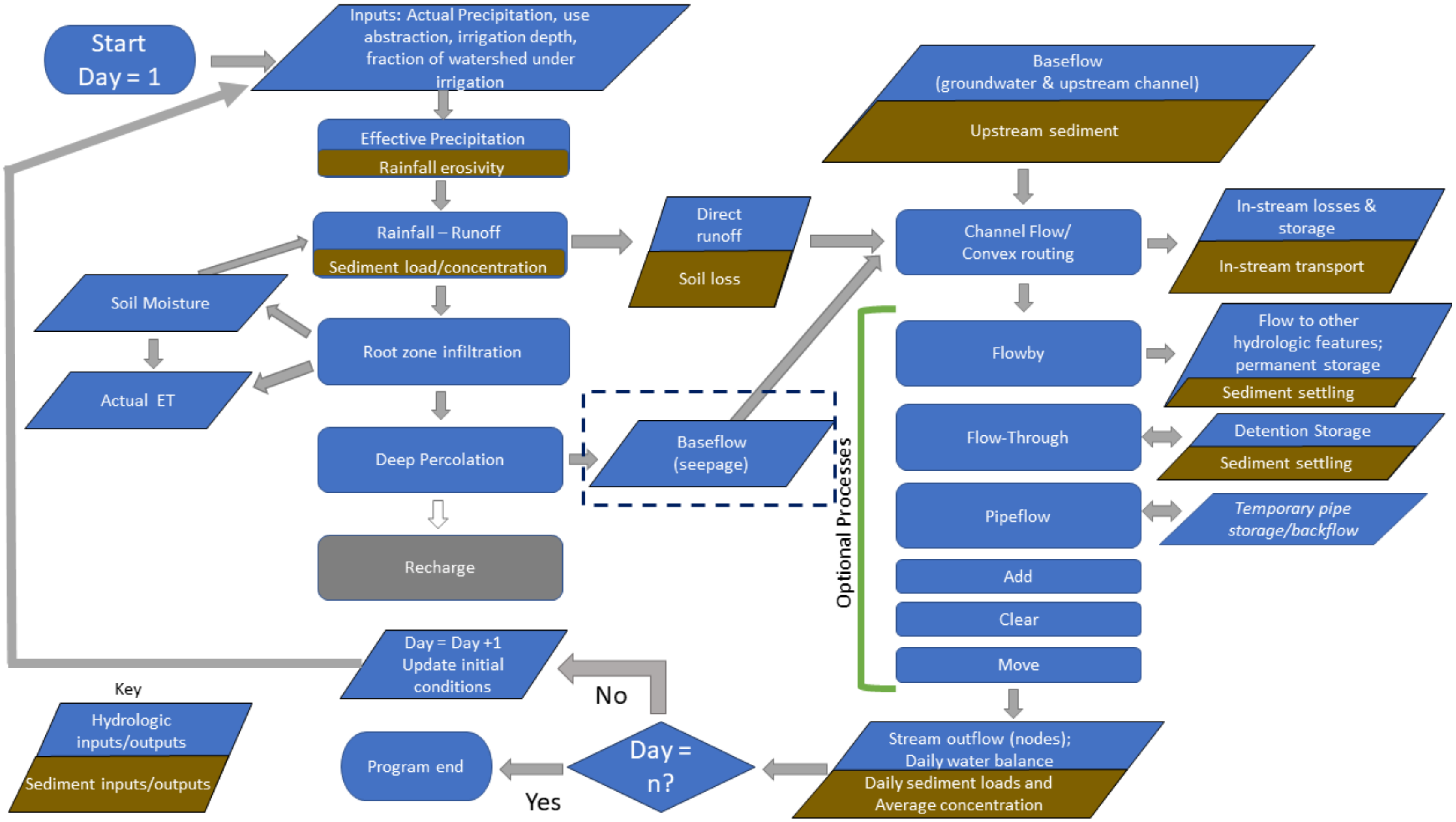

Figure 2-2. CUENCA link and node hydrologic model flowchart, detailing inputs (parallelogram), processes (rectangle), internal data storage (paralellogram), and outputs (parallelogram). Blue shapes contain hydrological components and in red are sediment transport components. The baseflow component is further developed in Fig. 2-2.

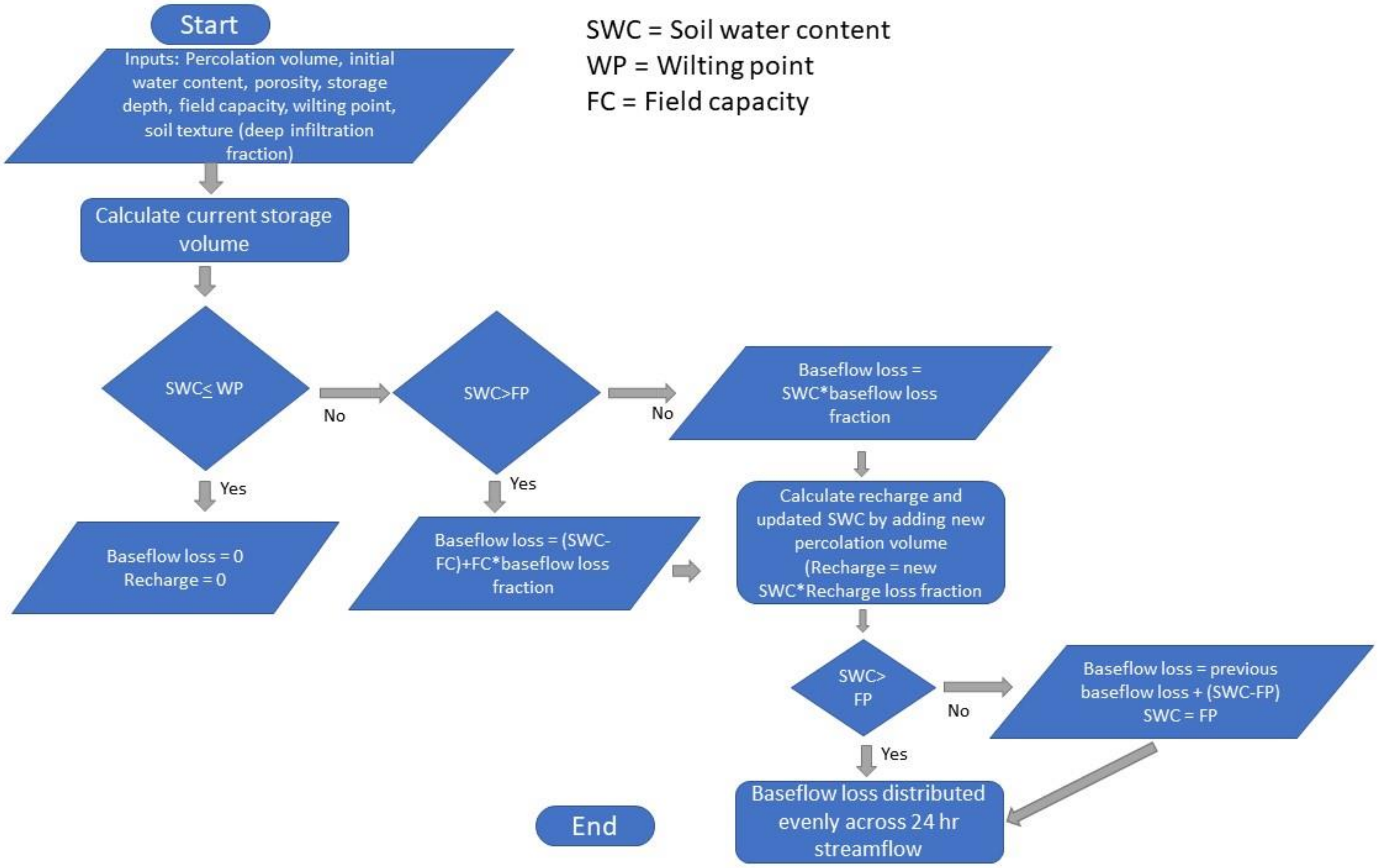

Figure 2-3. Deep percolation process within CUENCA. Shows partitioning to baseflow based on soil water content, wilting point, and field capacity.

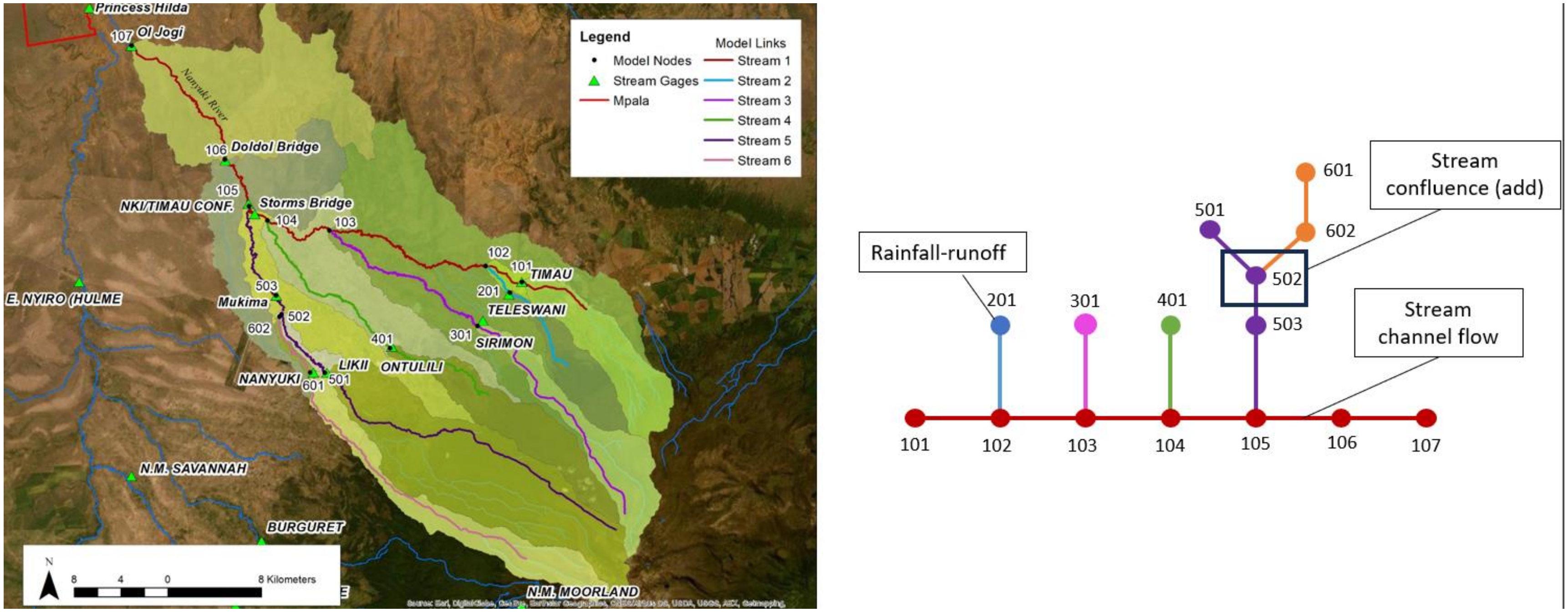

Figure 2-4. An example of a watershed converted to a link and node system for CUENCA model simulations. In this example, all nodes have a contributing watershed and require a rainfall-runoff process. All nodes except the first in each stream require convex channel routing. Where two streams meet, an add process is also required.

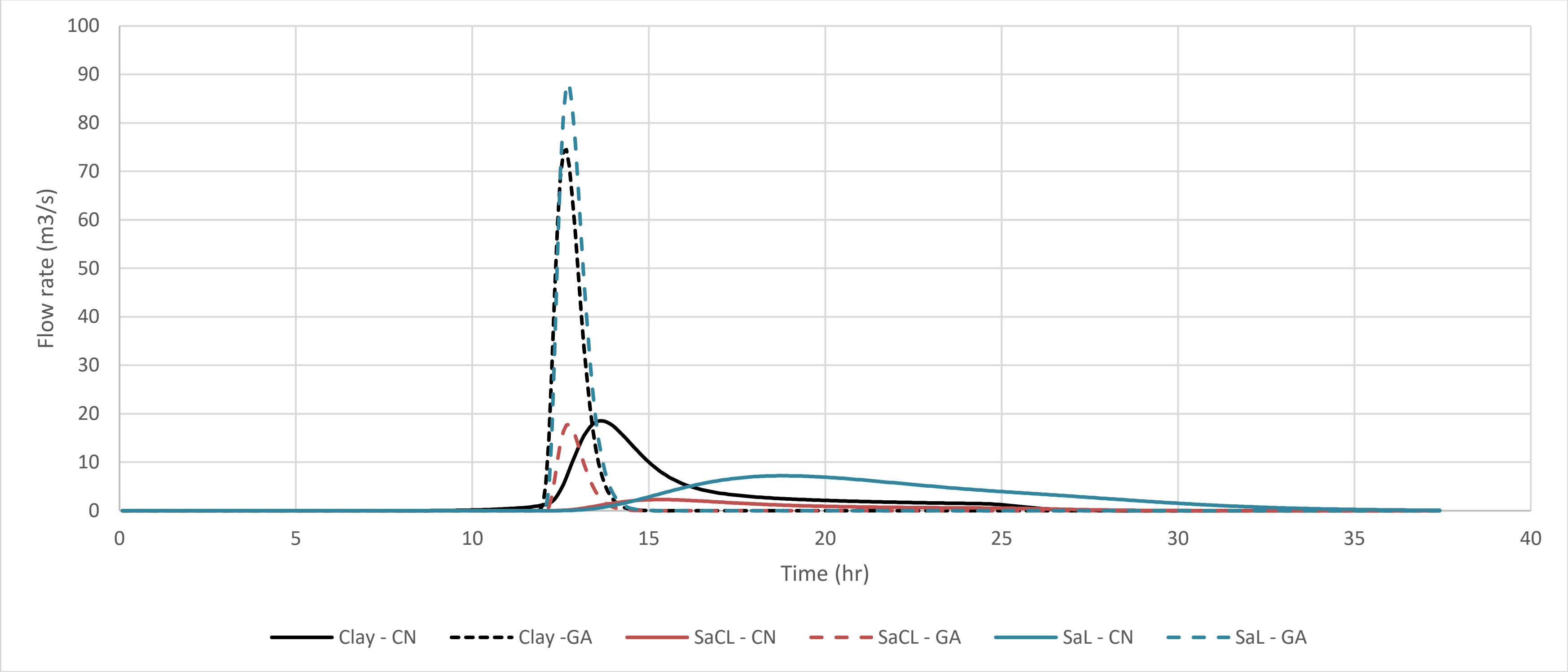

Figure 2-5. Comparison of CN and GA rainfall runoff outputs for a single event under clay, sandy clay loam (SaCL), and sandy loam (SaL) soil textures. Input values for each scenario are contained in Table 2-2. Runoff volumes correspond to those in Table 2-3, and are equivalent for the same soil textures.

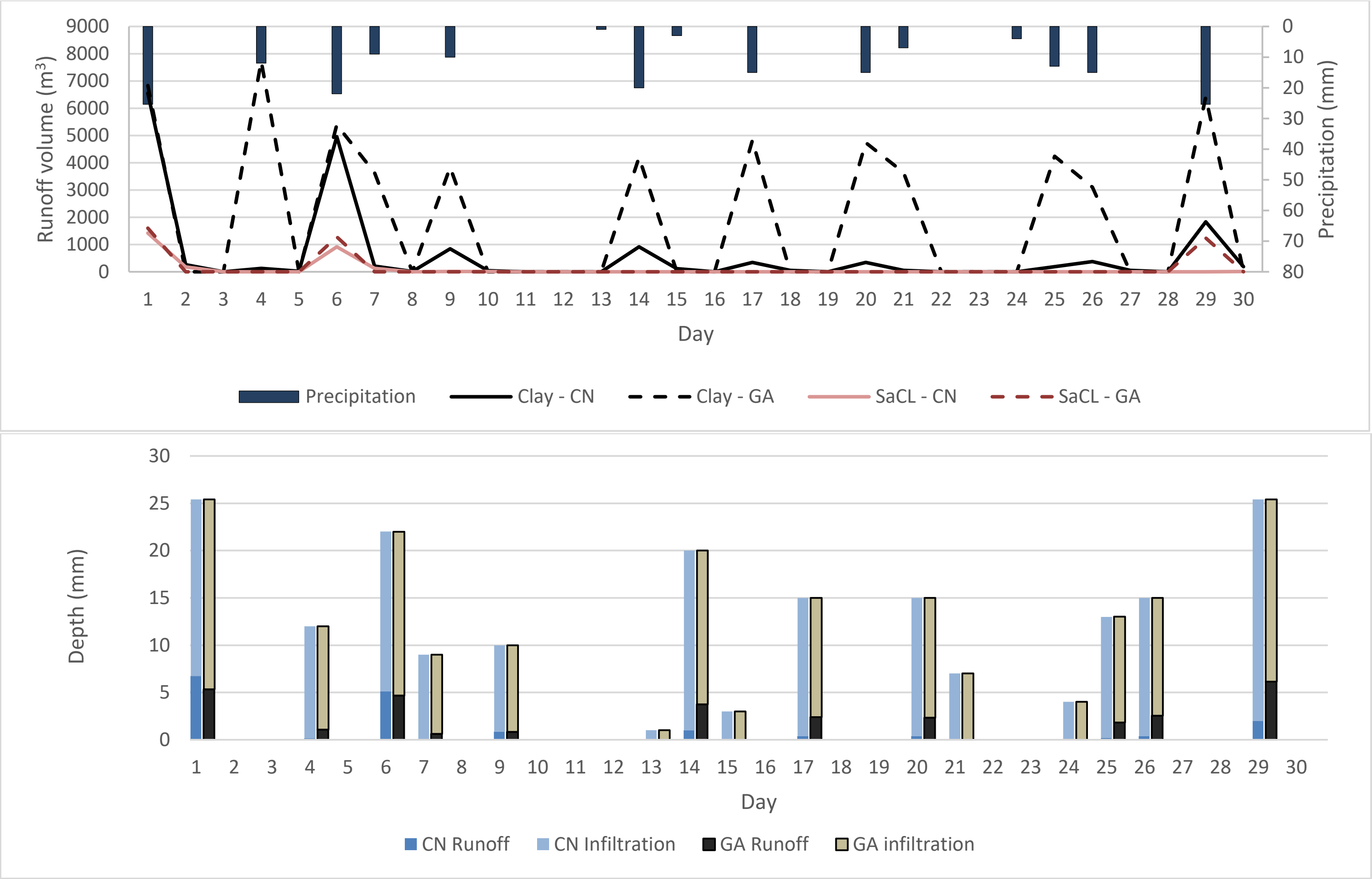

Figure 2-6. Comparison of CN and GA processes for A) (top) direct surface runoff entering the stream over 30 days of variable rainfall (all events <30 mm) under clay and sandy clay loam (SaCL) soil textures, and B) (bottom) runoff and infiltration volumes (equal to total precipitation on y-axis) during each rainfall event for clay soil scenario. Note that the top figure here refers to daily runoff volume and data points are at the daily time step.

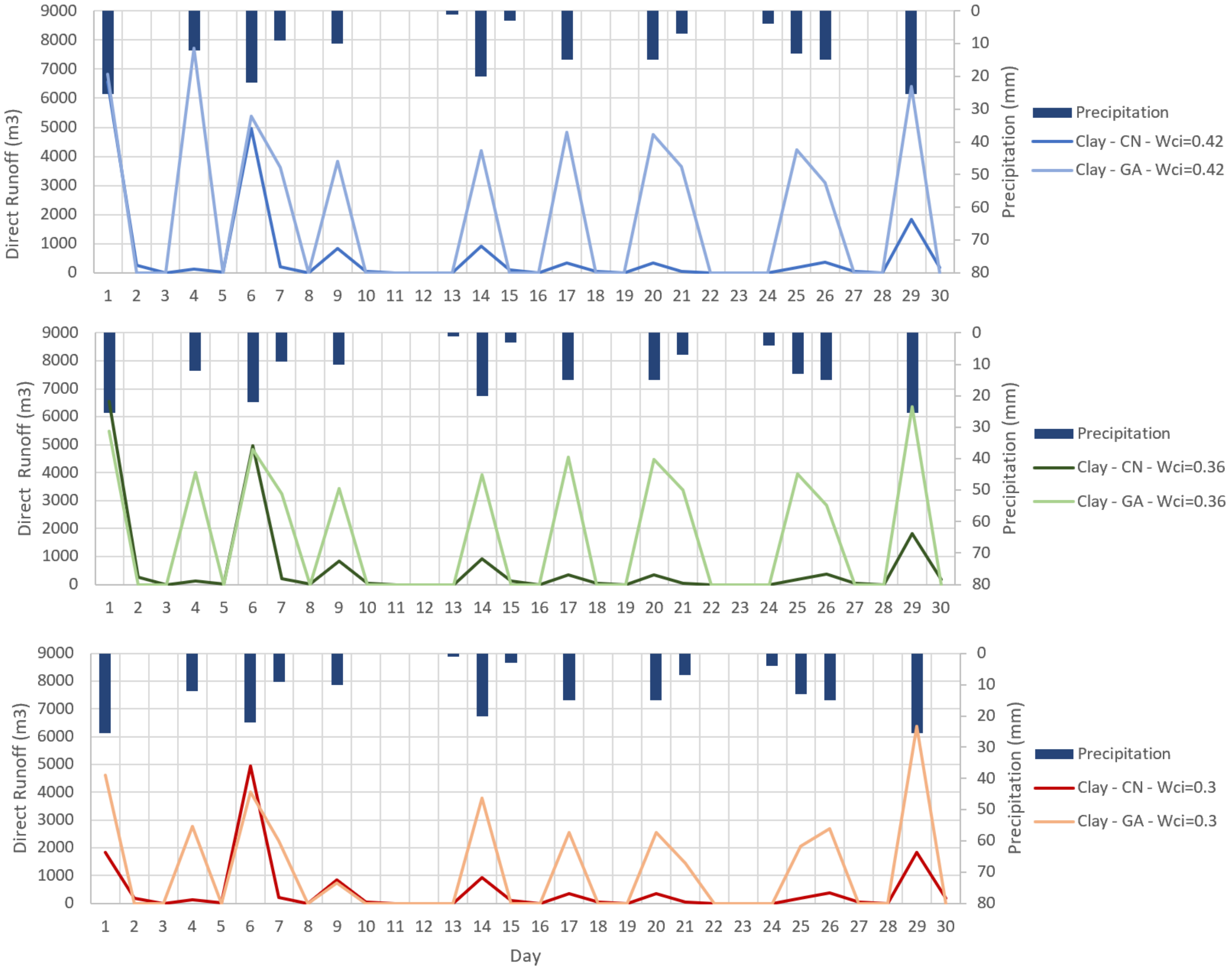

Figure 2-7. Comparison of direct runoff volume under different initial soil moisture conditions for conventional tillage clay scenario inputs in Table 2-4. Results show that initial conditions affect runoff volume for GA method, but not CN method.

CUENCA has been more rigorously tested on clay soils in anticipation of application in Laikipia, Kenya, which contains primarily clay soils.

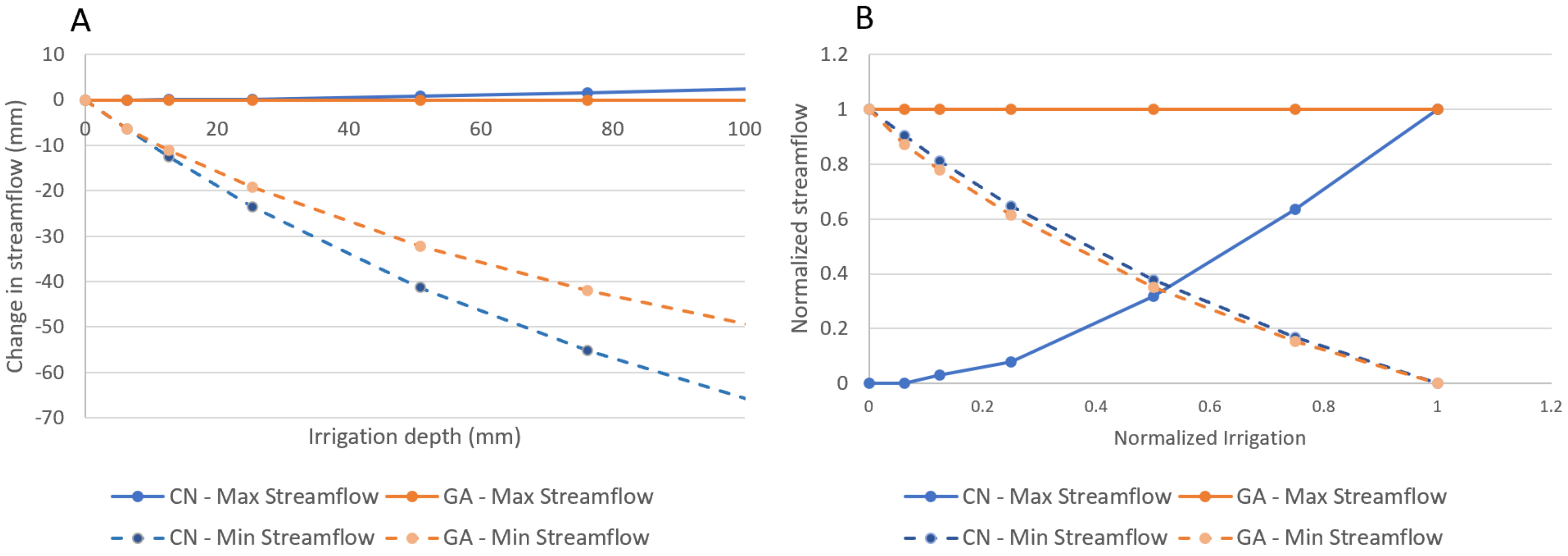

Figure 2-8. Sensitivity of streamflow to varying irrigation depths. Total change in streamflow depth is shown on the left (A), and normalized changes in streamflow minimum and maximums are shown on the right (B). Max streamflow refers to changes in the maximum observed value over the entire time-series of the simulation, while min streamflow refers to changes in the lowest observed streamflow during the simulation.

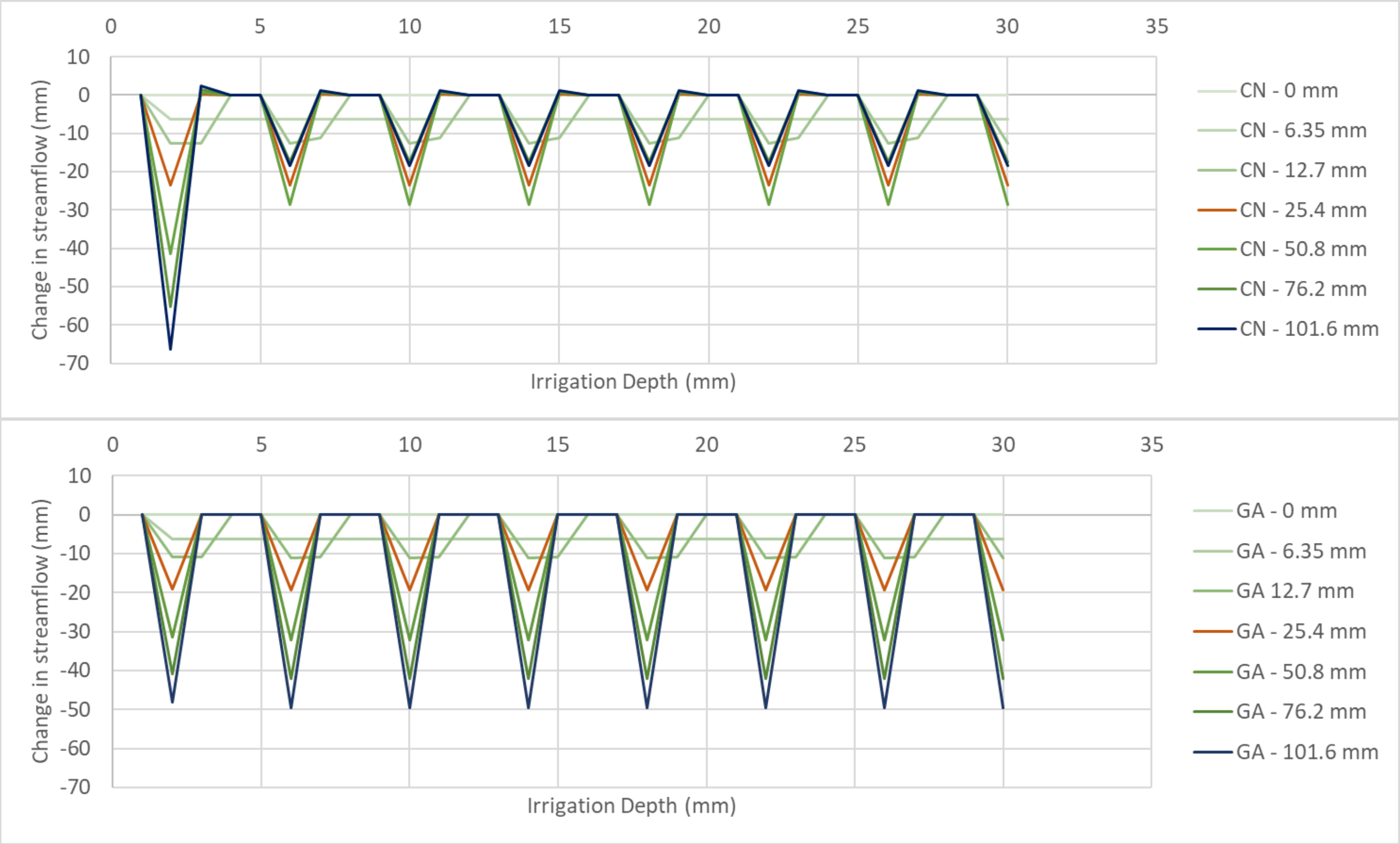

Figure 2-9. Time series of changes in streamflow (mm) based on different levels of irrigation and rainfall-runoff algorithms. calculation process. The top graph depicts the Curve Number method and the bottom graph depicts the Green-Ampt method. The GA method has much larger variation in streamflow, especially for deeper irrigation depths. This could be due to responses to soil moisture or uncertainty associated with assuming GA parameters based on CN runoff volumes.

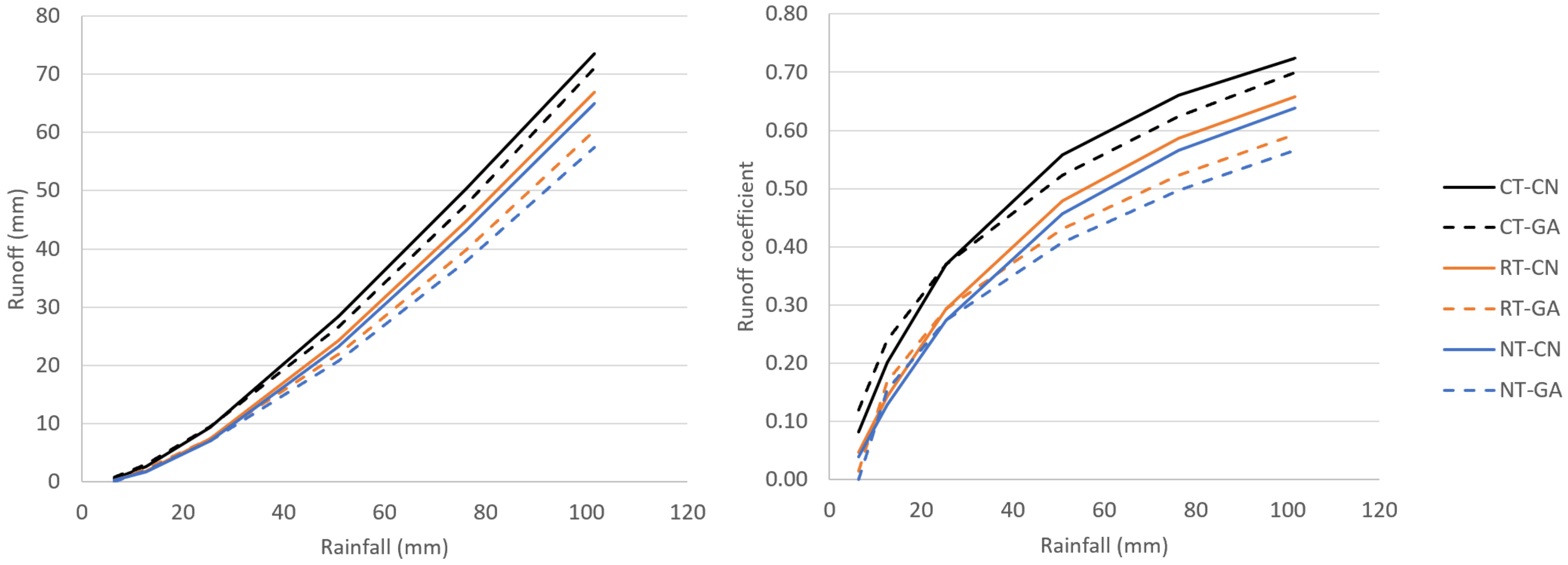

Figure 2-10. Direct runoff depths (left) and runoff coefficients (i.e. RO/P) (right) corresponding to different tillage types (CT=conservation tillage, RT = reduced tillage, and NT = no-till) under CN and GA model processes. For an equivalent depth of rainfall, CN has more RO during rain events larger than 25.4 mm (1-inch).

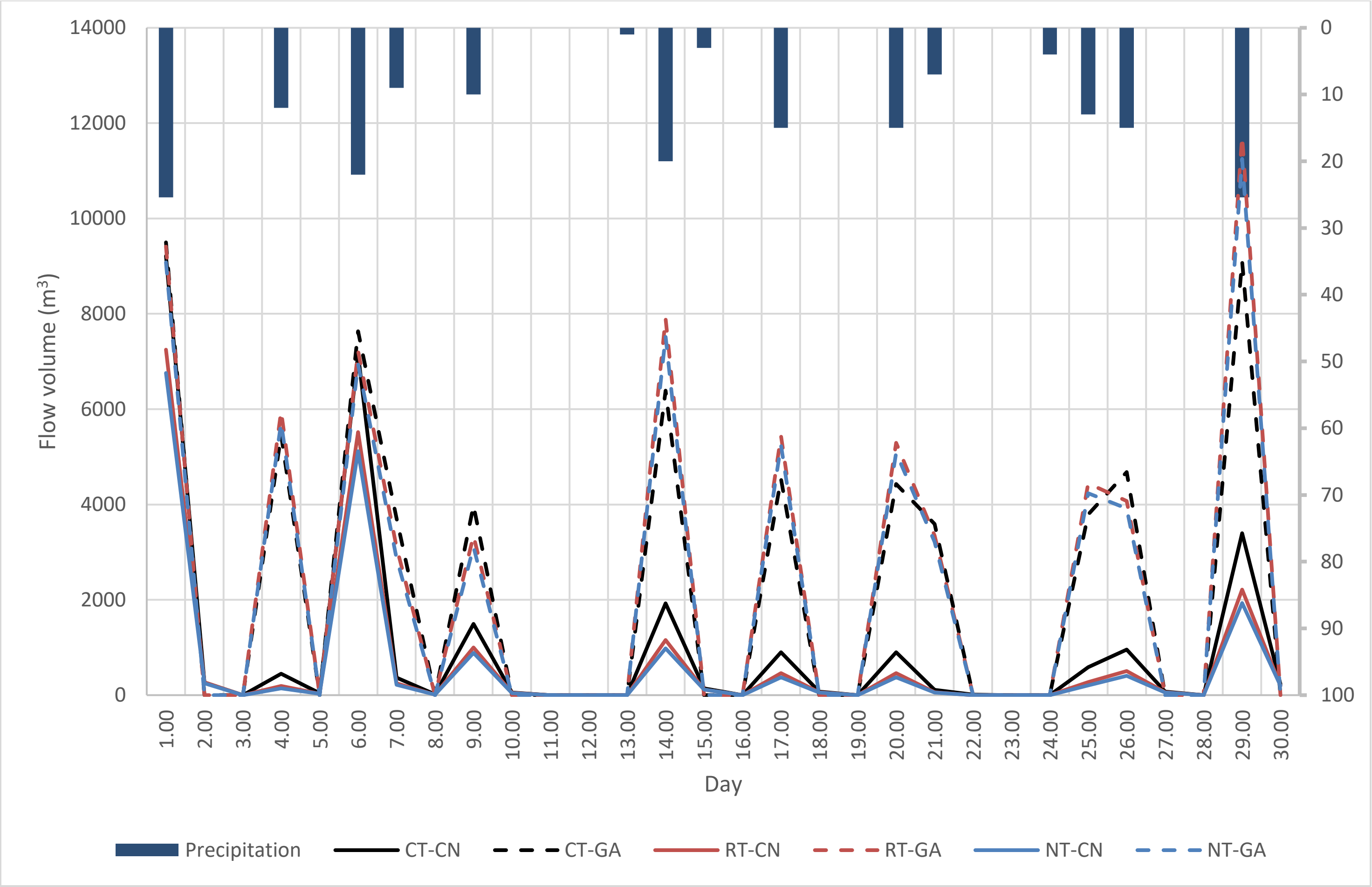

Figure 2-11. Results of a 30-day simulation comparing conservation tillage (CT), reduced tillage (RT), and no-till (NT) flow differences based on the GA or CN process. Overall the GA algorithm typically has more runoff. The water balance for this test scenario is contained in Table 2-4.

CHAPTER 3

IMPACTS OF REDUCED TILLAGE AND EFFICIENT IRRIGATION ALONG A STEEP RAINFALL GRADIENT IN LAIKIPIA, KENYA

As world population increases, agricultural production must increase to meet demand, and particularly in food insecure regions. Globally, just over 30% of food is produced on smallholder farms of less than 2 ha (Ricciardi et al., 2018), and in developing countries over 70% of food is produced on farms less than 5 ha (Samberg et al., 2016). Smallholder farms in Sub-Saharan Africa face many barriers to increased production, including lack of financial capital and infrastructure to invest in improved inputs, lack of market access, and insecure land tenure (Salami et al., 2010).

Increased food production can be achieved through agricultural extensification or intensification. During extensification, land area under cultivation increases, typically at the expense of natural ecosystems, and input levels of water, nutrients, and labor typically remain the same per land unit area as currently farmed areas. During intensification, agricultural production is increased on existing agricultural land by increasing inputs per land unit area and investing in technologies to improve crop yields, such as irrigation, fertilization or seed varieties. The increased use of inputs during intensification can result in degradation of natural resources through over-abstraction of water, soil depletion and erosion, and agrichemical pollution of waterways through runoff and groundwater leaching. In contrast, sustainable agricultural intensification (SAI) aims to produce more food per unit of land, preserve important ecosystem services, and provide resilience to system shocks and stresses (Pretty et al., 2011).

Existing (SAI) literature has provided conceptual frameworks to balance the problems of food production and natural resource management on large-scale and smallholder farms. These solutions include conservation agriculture practices, integrated soil fertility management, rainwater harvesting, and drip irrigation (Pretty et al., 2011; Rockstrom et al., 2017).

Conservation agriculture in particular has been studied in detail and comprises three practices: conservation tillage for minimal soil disturbance via no-tillage or reduced tillage, crop rotation and diversification, and permanent soil cover (FAO, 2020b). The combination of these practices should theoretically reduce soil erosion and chemical loss, increase soil water storage, enhance carbon sequestration, reduce the need for weeding, and improve economic profitability via increased yields and reduced labor costs (Thierfelder et al., 2018). However, for smallholder farms in Sub-Saharan Africa, results have been mixed. While few field studies analyze all of the above dynamics, several do measure yields and soil moisture under varying combinations of conservation agriculture practices. Some studies have observed increased yields, increased water use efficiency and soil moisture (Rockstrom et al., 2009; Fan et al., 2012; Thierfelder et al., 2015), contrasted with some that show no increased yields or soil moisture storage (Giller et al., 2009; Mupangwa et al., 2007; Pittelkow et al., 2015). There is evidence that field practices must continue for at least five years to support biophysical processes needed before a trend of increased yields is established (Thierfelder et al., 2015).

Most studies on conservation agriculture and other SAI practices have focused on field-scale impacts on crop yields, soil physical properties, and input use efficiencies (Cui, 2018; Sithole et al., 2019). Several studies have scaled up the impacts of intensification on hydrology and ecosystem services at the subwatershed level (Ngigi, 2008), although most focus on simple assumptions of landuse change. Due to the complex underlying landscape properties across the savanna ecological gradient (Good and Caylor, 2011; Campo-Bescos et al., 2013), and processes that both influence and are impacted by agricultural production, there is a critical need for a landscape scale assessment of the impacts of SAI practices on downstream natural resources.

Conversion from conventional (i.e. flood or furrow) irrigation to efficient irrigation (i.e. drip) is often promoted as a water-saving technique and a requirement to achieve food production goals amid increasing pressures on water resources (Jagermeyr et al., 2015). However, it is difficult to compare across studies due to differences in water-saving measurements and definitions (van der Kooij et al., 2013), and improved water use efficiency at the plot-level often does not scale up to the watershed either due to subsequent increases in irrigated area or changes in partitioning to evapotranspiration (Grafton et al., 2018; Scott et al., 2013). In a Mediterranean climate, Pool et al., (2022) identified similar groundwater recharge under drip and conventional irrigation scenarios in wet years and reduced recharge under drip irrigation scenarios in dry years. Therefore, a similar assumption could be made that drip irrigation would have less negative impacts at the watershed scale under wetter climatic conditions. The source and storage of irrigation water also impacts watershed scale sustainability, since rainwater harvesting with subsequent irrigation can reduce peak stormflows and redistribute water across the landscape when it is most needed (Baker et al., 2012).

Conversion from conventional tillage to reduced or no-till is typically assumed to increase soil water-holding capacity and saturated hydraulic conductivity (Ksat) through increased residue retention and improved soil structure, as well as lack of a hard pan created through tillage implements (Verhulst et al., 2010). However, results on changes in Ksat have been mixed (Strudley et al., 2008; Blanco-Canqui and Ruis, 2018). Fuentes et al. (2004) observed similar Ksat values in prairie soils under conventional and no-till systems, and these values were one order of magnitude less than Ksat values in native, undisturbed soils even after 27 years of no-till. Results from across various hydro-climatic regions indicate that Ksat impacts are most likely related to rainfall intensity, ET, and soil texture (Tomer and Locke, 2011; Bosch et al., 2012; Didone et al., 2014; Endale et al., 2014; Easwaran et al., 2021). Ksat is difficult to measure in both the field and the laboratory, which may contribute to conflicting measurements (Fodor et al., 2011; Bagarello et al., 2006; Reynolds et al., 2000). Bagarello et al. (2021) determined that 10-20 Ksat measurements for a plot of size 44 m^2 are needed to characterize clay soils while accounting for uncertainty.

Landscape-level studies typically focus on large, homogenous agricultural landscapes in food secure nations, where large machinery is the primary agricultural tool (Bowmer, 2011). In addition, there is typically easier access to existing data and better infrastructure to collect new data, meaning that biophysical models can be more easily calibrated and validated. In Sub-Saharan Africa (SSA), where heterogeneous smallholder farms prevail, biophysical models will inherently be more complex to apply as they will need data with a high spatial resolution, where in most cases these data are unavailable or unvalidated (Ndomba et al., 2008; Vigerstol and Aukema, 2011). Therefore, these models may contain more uncertainty due to data inputs. In hydrologic models that may be used for management decisions, this increased uncertainty must be clear and transparent to increase stakeholder buy-in so that model results are actually considered in policy decisions (Voinov and Gaddis, 2008).

Laikipia county, in the savanna landscape of central Kenya, is a representative example of the global need to increase food production while preserving natural resources amidst changing climate and tenuous socio-economic circumstances. The county prioritizes agriculture and the environment for economic growth, listing horticulture, cereals production, livestock and tourism (i.e. wildlife tourism) as its four economic pillars (Laikipia, 2020a). These pillars coincide with the main agricultural actors in area: large mechanized farmers on the high plateau around Mount Kenya, producing cereal and legume crops, horticulture and floriculture; medium and small scale farms, focusing on horticulture and some staple crops; small mixed farms with horticulture, staple crops and livestock on the semi-arid plateau; and large livestock ranches that double as conservancies in the downstream savanna.. The area is also subject to increasingly more variable rainfall and a limited river water supply that must be managed in the high- and mid-stream regions for agricultural production and household use, and downstream to support savanna wildlife and livestock (Wiesmann et al, 2000; Gower et al. 2016; Lanari et al. 2018). To work towards increased food production during the dry season, the county is promoting large-scale commercial irrigation and expansion of rainwater harvesting for existing horticultural crops with the simultaneous diversification to drought-resistant cereals, legumes, and horticultural crops (Laikipia, 2020a). The large-scale farms converting to drought-resistant crops have also begun to implement conservation tillage, crop rotations, and soil coverage as part of SAI. In addition, much of the livestock in Laikipia grazes alongside wildlife on large conservation ranches in downstream savanna regions (Georgiadis et al., 2003), intertwining the success of livestock and wildlife tourism. With a steep rainfall gradient and the presence of both volcanic soils and expanding clay soils, Laikpia is a very heterogeneous watershed and blanket recommendations for SAI practice adoption may not be practical to meet the region's intended development goals.

We hypothesize that in a region such as Laikipia, with a steep landscape and rainfall gradient, different SAI practices are preferable at different location along the rainfall gradient to reduce drought frequency downstream. Specifically, drip irrigation utilized in the higher (>950 mm/year) region and reduced or no-till in the intermediate rainfall (between 700 and 950 mm/year) region will strike a balance between reduced water consumption in the region where there is enough water for irrigation and improved soil water storage in the region that may be more susceptible to drought. To test this hypothesis, we use the simplified CUENCA process-based hydrologic model that is sensitive to parameters affected by SAI processes. In this study two years of streamflow and rainfall data were collected in an extensive monitoring network installed in the region and were used to model. SAI adoption scenarios of different intensity and location in the region were evaluated in terms of streamflow changes at different point in the watershed.

3.1 Methods

A summary of the methods for this chapter are contained in Figure 3-1. First, details about Laikipia County are shared, as well as field gauging information. Then, scenarios were developed to test different tillage and irrigation intensities. These were evaluated using changes in total flow and flow minimum and maximums.

3.1.1 Study Area

Laikipia County, Kenya (Figure 3-2) is located on the north/northeast side of Mount Kenya. The water in this region flows northwest to the Ewaso N'giro River, where savanna wildlife and humans depend on the river. Historically fed by the glaciers of Mount Kenya, glacier recession in the last century has reduced the steady supply of water, and the increased human population in the region has increased pressure on water resources (Prinz et al., 2018). The Nanyuki River is a branch of the Ewaso N'giro that flows through Nanyuki Town and is the primary water source for agriculture in the region. In this study, the Nanyuki watershed is selected for model evaluation. This watershed comprises the contributing area of Mount Kenya and stretches to the savanna, where ranchers and pastoralists share land with wildlife. This watershed (Figure 3-3) is 1,130 km^2 and contains a mixture of small, medium, and large farmers using a range of mechanized and non-mechanized agricultural tools. While large farms have boreholes and large irrigation ponds, many smallholder farmers rely on direct river water abstraction for crop irrigation.

Laikipia County has a bimodal rainfall distribution, with two rainy seasons from March to May and October to December. The prominent soils in the watershed are red clay (ferric Luvisols) and dark clay soils with vertic properties (verto-luvic Phaezems) (Liniger, 1991), and volcanic soils were also observed during this study. These vertisols are commonly called “black cotton” soils, and have characteristics of high clay content, high shrink-swell and cracking capacity, and very low hydraulic conductivities.

3.1.2 Data

Field data collection. From July 2021 – July 2023, eight streamgages, five rain gauges, and two barometric pressure loggers were installed to record data in the Nanyuki watershed. Figure 3-4 shows the watershed as well as the location of the gauges. The locations were chosen based on both modeling considerations and safety for gauge location. For the model, we targeted stream reaches to verify the model mass balance (such as 2 out of 3 branches at the confluence of a stream), and attempted a distribution of rain gauges throughout the rainfall gradient. During location selection, we contacted local community members and chiefs to discuss the project and intended outcomes for community buy-in, which we also hoped would reduce potential vandalism of the streamgages. Rain gauges were placed on private property to reduce theft of solar panels, batteries, and copper wire. In several instances, streamgage construction required reinforcing local foot bridges for safety. During the study period, two gauges had to be rebuilt, one due to a large storm and another due to a car accident.

RainWise tipping bucket rain gauges (Rainwise, 5% accuracy at 2” per hour, resolution 0.1 mm/hr, and range 0-7.8 in/hr) collected 5-minute resolution data using a Campbell CR1000data logger. High resolution 5-minute streamflow data was collected from August 2021 to July 2023 at eight streamgages in Laikipia Kenya. A Solinst Levelogger 5 was used at each gauge to collect water height and subsequently corrected with barometric pressure data. Figure 3-5 shows examples of the rain gauge and Levelogger in the field, and Appendix B contains site photos of each stream gauge.

Rating curves (Appendix C) were developed to convert flow height in meters to flow rate in cubic meters per second. A topographic survey was also conducted to georeferenced the monitoring instruments and verify the hydraulic gradient across the landscape.

Remote sensing data collection. Table 3-1 provides a summary of remote sensing hydrometerological data used in the input files. Due to the steep rainfall gradient, precipitation data was also acquired from the Climate Hazards Group InfraRed Precipitation with Station data (CHIRPS) for June 2021 – June 2023 (Funk et al., 2014).

Temperature data were obtained from MODIS AQUA Land Surface Temperature, LST and 3-band Emissivity Daily Global 1km dataset (MYD21A1D and MYD21A1N versions 061) (Hulley, 2021a; Hulley, 2021b) . The AQUA mission collects LST daily at 1:30 am and 1:30 pm local times, which can be used to approximate daily maximum and minimum temperatures.

Potential evapotranspiration data was obtained from the MODIS Terra Net Evapotranspiration 8-Day Global 500m (MOD16A2 version 061) and interpolated to a 1-day temporal resolution.

Wind speed data were obtained from the MERRA-2 M2T1NXFLX: Surface Flux Diagnostics V5.12.4 product and corrected using a logarithmic wind speed height correction algorithm included in FAO-56. The MERRA-2 product provides surface wind speed measured in m/s at approximately 60m above ground level, and the data included in this dataset were corrected to 2 m height using this algorithm (FAO, 1998).

Landcover data were obtained from the ESA WorldCover 10 m resolution landcover map (Zanaga et al., 2021), based on Sentinel-1 and Sentinel-2 data. This map contains 11 landcover classes, and has a 76.7% accuracy. Despite this, some discrepancies were noticed in agricultural landcover. Therefore, large farms were hand-delineated in ESRI ArcMap and a high resolution smallholder agricultural landcover was provided by The Nature Conservancy, who were collaborators and funders for this project.

Elevation data were acquired from the CGIAR Consortium for Spatial Information (CGIAR-CSI) Shuttle Rocket Topography Mission (SRTM) digital elevation model at 3 arc second (approximately 90 m) resolution is included in the database (Reuter et al., 2014). This product has been corrected to fill voids, making it more useful for hydrologic modeling. The vertical error of the dataset is no more than 16 m.

Soil physical and chemical properties were obtained from the ISRIC AfSis project that developed the 250 m SoilGrids dataset using legacy soil profiles, sentinel sites, and environmental covariates (Hengl et al., 2015). The sentinel sites included 60 10km-by-10km sites that were sampled at a high spatial density (16 sampling clusters, 10 sampling plots per cluster, and 2 depths per cluster for a total of 320 unique samples per sentinel site).

Environmental covariates included for model parametrization include: 250 m MODIS products (Mid-infrared Reflectance Band 7, and Enhanced Vegetation Index (EVI) Long-Term and Monthly Averages (MOD13Q1); SRTM DEM v 4.1 elevation, slope, and SAGA GIS Topographic Wetness Index (TWI) at 250m; GlobeLand30 2010 landcover map resampled to 250 m from 30 m; and the SoilGrids 1km product downscaled to 250 m. The SoilGrids product contains physical and chemical measurements at 6 depths in a soil profile. These include texture (sand, silt, and clay fractions), organic carbon, pH, volume of coarse fragments, bulk density, cation exchange capacity, total nitrogen, extractable aluminum, exchangeable acidity, exchangeable calcium, exchangeable potassium, exchangeable magnesium, exchangeable sodium, and sum of exchangeable bases.

Hydrologic soil groups for the region were obtained from the HYSOGs 250 m product was developed based on the 250 m SoilGrids product using textures and depth to bedrock. This product can be used for curve-number based runoff models.

3.1.3 Uncalibrated Model Parametrization and Evaluation

Model input preparation. The Nanyuki Watershed was first organized into a nodal network as required by the CUENCA link and node model. The first nodes were decided based on stream gauge locations, and then based on confluence locations and where necessary to reduce subwatershed size. Figure 3-3 shows the final watershed configuration and node map containing 50 nodes. Across these 50 nodes, landscape and stream parameters were lumped to a single average value representative of typical conditions of the subwatershed contributing to the node. Curve number was estimated based on a combination of landuse landcover maps, hydrologic soil groups, and knowledge of local landscape conditions. Green-Ampt parameters (Ksat, Sav, Theta S) were estimated using soil texture from AfSis soils data and parameter transfer functions from Saxton (2006). Observed streamflow and rainfall data were summed to daily flow volume (m^3) and rainfall depth (mm) used in the model inputs.

Uncalibrated model parametrization. A simple direct parametrization was followed using inputs derived from the field and other sources in the database. Under this uncalibrated condition the model was evaluated against different options of existing data (precipitation products) and alternative processes (using both the Green-Ampt and Curve Number methods). Other factors like surface storage/ponding, and depth of subsoil layer were identified using a subset of the watershed (the Likii Branch of the Nanyuki River) which has high gauge density and flows from areas of high rainfall to intermediate rainfall. This helped to evaluate of data accuracy, particularly rainfall, because streamflow data could be used directly as an input and the CHIRPS rainfall product could be evaluated against field observed rainfall. Because many models are particularly sensitive to precipitation, groundwater/baseflow dynamics, and curve number, a conservative approach, using parameters exactly as they were calculated using subbasin area-based averaging, was conducted. These sub-basin directly calculated values were also checked against available field data to assess whether they were realistic for the region.

3.1.4 Model Application: SAI Scenarios

Baseline scenarios were conducted using the inputs as determined by subwatershed calculations. These baseline scenarios were compared to observed data to determine whether the Curve Number method or Green-Ampt method provided to best fit.

Irrigation scenarios were developed for typical irrigation (irrigate 25.4 mm if 25.4 mm of rainfall has not occurred in the prior 4 days) and drip irrigation (irrigate 6.35 mm if 25.4 mm of rainfall has not occurred in the prior 4 days).

Tillage scenarios were developed by adjusting the NRCS curve number based on literature recommendations for each tillage practice (Rawls et al., 1980; Sur et al., in progress), with a 9% reduction in curve number for no-till and a 7% reduction for reduced tillage. Equivalent Green-Ampt values for saturated hydraulic conductivity and suction at the wetting front under a 25.4 mm rain event were determined to match the streamflow and mass balance components from the CN simulations adjusted for tillage. This ensures consistency between both methods for comparison.

SAI adoption scenarios were simulated across the landscape gradient by dividing the watershed into “high” and “intermediate” rainfall zones (see Figure 3-6). Each scenario was tested assuming 100% of agricultural land in a zone adopted the practice. Table 3-1 contains a summary of the scenarios. Table D-1 contains irrigation fractions for each irrigation scenario. Table D-2 contains a summary of agricultural landcover in each subwatershed and adjusted tillage parameters. Figure 3-6 shows the rainfall areas and agricultural landuse.

The results of each scenario were evaluated for local streamflow dynamics at each gaged node. Drought duration and frequency as well as peak flows were evaluated for each scenario adoption level and watershed location to determine whether adoption location significantly impacts extreme streamflow dynamics.

3.2 Results

The results from the Laikipia case study are divided into model evaluation results, where CN and GA algorithm predictions as well as different precipitation datasets are evaluated. Then, results from irrigation scenarios are discussed. Finally, results of tillage scenarios are discussed.

3.2.1 Model Evaluation

Precipitation was observed to have an important impact on model outcomes, and both CN and GA simulations contained peaks with rapid recessions after June 2022. Therefore, three datasets were developed for testing impact of rainfall on model predictions. The "all observed" dataset utilizes rain gauge data with simple spatial statistics to adjust across the Laikipia watershed. The "ensemble CHIRPS" dataset uses primarily CHIRPS data with observed data used only in the subwatersheds where it was observed. The "ensemble observed" dataset includes primarily observed data with CHIRPS data substituted when storm events were clearly missed (particularly for upper elevations of Mount Kenya) by rain gauges. The "ensemble observed" dataset performed best and was thus used in subsequent analyses. Figure 3-7 shows a comparison of these three different rainfall inputs, and Table 3-4 shows the equivalent NSE values.

Figures 3-8 through 3-14 show the results of CN vs GA model simulations plotted against observed data for the corresponding stream gauges. The CN simulation had an NSE of -1.86 and the GA simulation had an NSE of -0.57. These values are both poor, and advanced (inverse) calibration of the model could be used to improve predictions. Additionally, there is a distinct difference in model performance before and after June 2022. Table 3-3 contains the NSE and RMSE values for each streamgage in Laikipia (note that the Likii gauge (node 704) was used as a data input, and therefore has an NSE=1 and low RMSE), for the total study period, then separated before June 2022 and after June 2022. Based on classifications by Ritter and Munoz-Carpena (2013), at the Juakali gauge performs good to very good with an NSE of 0.83 and 0.92 for CN and GA algorithms, respectively.

Figure 3-10 shows node 704, the Likii streamgage, which matches observed data perfectly because it was used as an input for the simulation. In general, the GA method underestimates streamflow, while the CN method overestimates peak flows but still has a quick recession. Neither process maintains consistent river flow as is shown in the observed data. This may partially be due to poor quality precipitation data; many small, consistent events on Mount Kenya would most likely maintain more consistent flow. It could also be attributed to higher springflow from Mount Kenya or irrigation dynamics. The peaks from the CN method could be adjusted using more landscape storage. Many large farms do have irrigation ponds that hold a significant amount of water.

While calibration could improve the model simulations, the ability of the uncalibrated model to reproduce the observed streamflow dynamics, particularly with the GA method (higher NSE), support its application to evaluate the SAI scenarios. Although the GA method had a better NSE, there is direct literature to support adjustment of CN values for tillage scenarios.

3.2.2 Irrigation Scenario Results

Figure 3-15 shows the results of all irrigation scenarios at node 116, which is just downstream of the high rainfall area; node 117, which is just downstream of the intermediate rainfall area, and node 119, which is in the savanna. Table 3-5 includes changes in flow volume when compared to the baseline CN simulation over the entire study period, as well as maximum and minimum daily changes in flow volume. The table is organized by observed changes at nodes 106, 117, and 119, which correspond to the high rainfall area, intermediate rainfall area, and savanna. Node 106 does contain some contributing area from the intermediate rainfall area, so changes are still observed at this node for scenarios M 25.4 and M 6.35, where irrigation is only altered in intermediate rainfall areas.

At the edge of the high rainfall area, the scenario that has the most impact on flow volume is applying conventional irrigation (25.4 mm) in the high rainfall zone. It reduces flow volume by 46% over the modelling period, and increases some daily flows up to 100%. When evaluated at the two downstream reaches, the total flow volume stays essentially the same at the intermediate reach (-0.01% change) and savanna (0.02% increase). However, it does seem to impact the flashiness of the savanna section, causing considerable increase in maximum flow (2110%) and 100% decrease in flow some days. This maximum flow increase occurs during low flow conditions, indicating that it is runoff as a result of return flows from drip irrigation.

At the intermediate rainfall area, drip irrigation applied at the intermediate rainfall area (M 6.35), conventional irrigation applied to 50% of the total agricultural area,(H&M 25.4) and drip irrigation applied to 50% of the entire agricultural area (H&M 6.35) all cause the largest decreases in flow volume, with 61.6%, 60.2%, and 61.6% reduction respectively. Conventional irrigation applied in the intermediate rainfall (M25.4) area is only slightly better, causing a reduction of 59.7% These correspond to minimal increases in daily flow, and 100% decreases in daily flows on some days. At the savanna region, these impacts decrease to flow reductions of approximately 29%, but the maximum daily flow observed increases by up to 160% for the first three scenarios and 785% for M 25.4. All four of these scenarios have the largest impact in the savanna region of the watershed. The drip irrigation scenario applied at the high rainfall area has the lowest impact on flows in the savanna region, barely affecting flow volume or daily maximum, although it does sometime reduce up to 100% of flow.

3.2.3 Tillage Scenario Results

Table 3-6 shows a summary of the changes in flow volume under no-till (NT) and reduced tillage (RT) in high rainfall areas (H) or intermediate rainfall areas (M), or both (H&M) compared to the CN baseline scenario in the Nanyuki River watershed. The H&M scenarios contain both 100% and 50% coverage of agricultural areas to provide a more comparable analysis on a “per area” basis to the high and medium rainfall areas only.

In the high rainfall area, 100% conversion to no-till and reduced till decreases flow volume by 20.7% and 17% respectively. This corresponds to daily flow increases of up to 4.8% and peak reductions up to 54.4% in the no-till scenario. When viewed across the landscape, these reductions are dampened, and flow volume is only decreased by about 2% at the intermediate rainfall area and 1% in the savanna region. Daily flow maximums increase to up to 11.6% and 7.4% at the intermediate and savanna areas. The increases in flow volume typically occur several days after a rain event, and can most likely be attributed to slight increases in baseflow on lower flow days as a result of increased soil moisture. No-till and reduced tillage at 50% coverage reduce flow volumes by 11.8% and 9.5% respectively, and still provide some modest (3.7% for NT and 2.1% for RT) increases in daily maximums as well as peak reductions up to 33%.

In the intermediate rainfall area, 100% conversion to no-till and reduced till in both high rainfall and intermediate rainfall areas contributes to a 7.3% reduction in flow, up to 75% increase in maximum daily flows, and peak reductions up to 46%. The NT-H, NT-H&M 50, RT-M, and RT H&M 50 scenarios perform similarly with between a 2-4% reduction in flow volume, although NT-H has lower changes in daily maximums and minimums, so it may not be preferred for baseflow resilience improvements.

At the savanna region, no-till across the entire watershed reduces total flow by 4.4%, the highest of all scenarios, but it also increases maximum flow the most at up to 264% and reduces peaks up to 72.5%. More analysis should be performed to determine what this means for flow timing, since the large flow increases on a percentage basis probably occur during low-flow times and do not correspond to significant increased flow volume. No-till at the intermediate rainfall area has the lowest flow volume reduction at 0.8%, but it also has one of the lowest maximum flow changes at only 6.5% and reduces peaks by up to 24.5%. At the far end of the watershed, the scenario converting to no-till broadly across 50% of the watershed might strike a good balance of flow volume reduction at 1.3% with larger increases in maximum flow change (258%) and possible peak reductions (-72.5%).

3.3 Discussion

Although the uncalibrated CUENCA link and node model did not provide a good fit for the Nanyuki River watershed, which was expected when using a simple direct parametrization method, the dynamics of the streamflow were properly matched. While more effective model inputs could be identified through inverse calibration to improve predictions, there is value in this simple uncalibrated model, since we know that the model is not overly influenced by fitting to groundwater flows or other estimated landscape parameters.

Table 3-7 contains a table comparing CUENCA with other process-based hydrologic models, including SWAT (Arnold and Fohrer, 2005), HEC-HMS (Fleming and Neary, 2004), and WaSiM-ETH (Schulla and Jasper, 2007), adapted from Haberlandt (2010). Haberlandt (2010) evaluated these models for their ability to be used as decision support systems. The author based criterion on spatial scale, temporal scale, degree of determination, target variables, complexity and handling, efficiency, performance, and sensitivity for a test case in a 1000 km2 in watershed in Germany. Although watershed configuration would affect the comparison between models, for this initial qualitative comparison we will assume the CUENCA test case in the Laikipia watershed (only 10% larger) is comparable. CUENCA is similar to these models, although still in its infancy stages. It may be most similar in processes and functioning to HEC-HMS, but HEC-HMS has high calibration requirements and a large number of calibration parameters. CUENCA has not yet been through an optimized calibration process or global sensitivity and uncertainty analysis, both of which are necessary to fully identify a niche for CUENCA within the hydrologic modelling landscape. It does not have the high-level of processes available that SWAT has, and will most likely not be sensitive to crop rotations as SWAT is, just based on the relatively simple data inputs for crop rotations available to CUENCA users. Currently, CUENCA fills a gap in the hydrologic modeling space due with both simple calibration requirements, simple inputs and easy handling, and flexibility for user specified watershed and infrastructure configurations. As the model is evaluated and optimized further, hopefully runtime is reduced and CUENCA can be used as a light model that is sensitive to agricultural management scenarios and landuse-landcover changes. Additionally, although all of the models in Table 3-7 are continuous daily simulations, literature indicates that SWAT is calibrated at the monthly scale, and may not provide accurate daily flow values to assess ecosystem function (Sudheer et al., 2007).

In this case, it is preferable to interpret the scenario outcomes as patterns rather than absolute values of streamflow change. The results of the irrigation scenarios confirmed previous work that observed reduced flows under high efficiency irrigation (Huffaker, 2008; Grafton et al., 2018). It also confirms the hypothesis that adoption of efficient irrigation practices is preferable in high rainfall zones, where local rainfall provides soil moisture and dampens any negative impacts on streamflow volume relative to abstraction water downstream. This can be compared with previous studies that observed no significant difference in recharge volumes during wet years (Pool et al., 2021).

The results of the tillage scenarios also corroborate previous work that observed reduced peak runoff and increased baseflow in soils under conservation tillage practices (Endale et al., 2014; Tomer et al., 2005). Bowmer (2011) discusses the difficulty of attributing changes in river flow to agricultural practices, and this is still a valid consideration. The CN method was developed in North American watersheds, and may not be the best model for evaluating tillage or irrigation practices in East Africa. Future use of the GA method for scenario analysis is necessary along with field-observed soils data to correct AfSis data as input values.

3.4 Conclusions

The CUENCA link and node uncalibrated model provides a poor fit for the Nanyuki River watershed, but the dynamics of the model results fit well with the observed seasonal dynamics. The Green-Ampt method exhibits a systematic error of underpredicting river flows, which makes it a good candiate for advance inverse calibration to improve the simulation results. However, the correct dynamics captured by the uncalibrated model supports its use in ungaged or data poor regions like maky in SSA and its use here for SSA scenario analysis. The results of the scenario analysis indicate that irrigation and tillage management may have optimal locations within the Nanyuki River watershed to maintain flows in the savanna region. Drip irrigation in the high rainfall area minimally impacts flows at Ol Jogi, and even conventional irrigation in this region performs better than drip irrigation in the medium rainfall area. On the other hand, all the tillage scenarios have relatively small impacts on flow volume once the river reaches the savanna. They all slightly decrease flow volume, but including reduced till or no-till management in the intermediate rainfall area at some level does provide the same amount or more baseflow downstream than utilizing it only in high-rainfall zones. This work represents a first step towards identifying hyper-local suitability for SAI practices in Laikipia and other tropical regions that transition to savanna.

The limitations of this work include accuracy of input data and uncertainty with modelling the physical environment. Field-measured saturated hydraulic conductivity was about one order of magnitude lower than those estimated using the AfSis data, which would significantly affect results. Many of the soils observed contained volcanic properties as well, which do not fit within the USDA textural analysis. Since the suction at the wetting front and the saturated hydraulic conductivity values were derived from estimates using USDA texture classifications, the Green-Ampt properties of volcanic soils should be used to parameterize the model in future works. Even with field-observed data, there is uncertainty associated with each measurement, as well as the subsequent laboratory analysis or conversion from stream water height to flow rate. Additionally, it is impossible to capture every process in the landscape in the model. Future work should include a comprehensive inverse model calibration and uncertainty analysis to ensure that the model results are not biased by certain input data as well as scenarios based on field-observed soil properties under no-till conditions. Since CUENCA is a (sub-)daily model, event-specific dynamics should be explored more to improve model performance, including runoff coefficients per storm event and typical storm duration and intensity.

This work can be used to guide decision makers when they need to identify best practices for water resource protection in Laikipia County specifically and provide possible insight for those in tropical watersheds that have a high rainfall gradient. Future work should continue to evaluate tradeoffs between practices (e.g. no-till reduces volume of streamflow ultimately, but it does provide more flow during dry periods) and identify benefits across the ecosystem. These scenarios should be coupled with a crop model to evaluate tradeoffs in crop production based on these practices. Since CUENCA is a daily streamflow model, it should be improved in Laikipia so that it can be used to evaluate the frequency of river drying at the savanna landscape for impacts on wildlife and livestock.

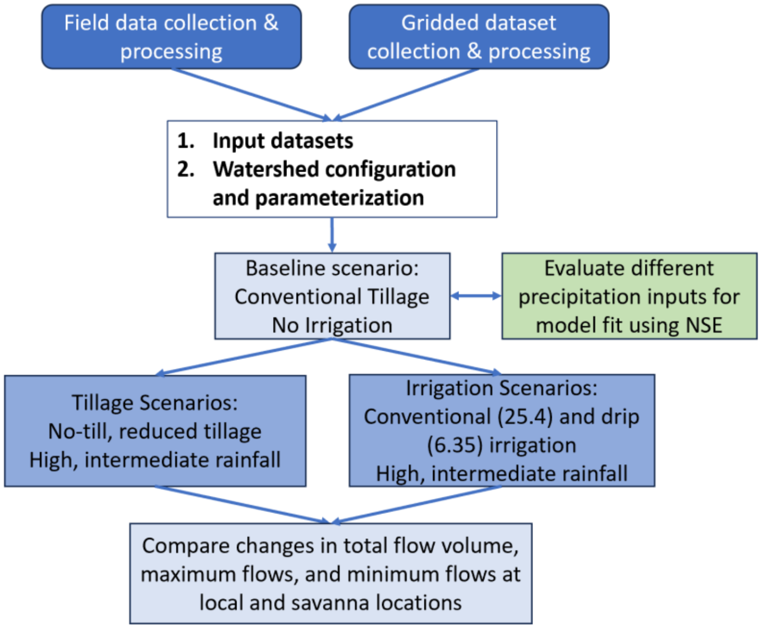

Figure 3-1. Methods for baseline and scenario evaluation for tillage and irrigation scenarios throughout Laikipia Watershed.

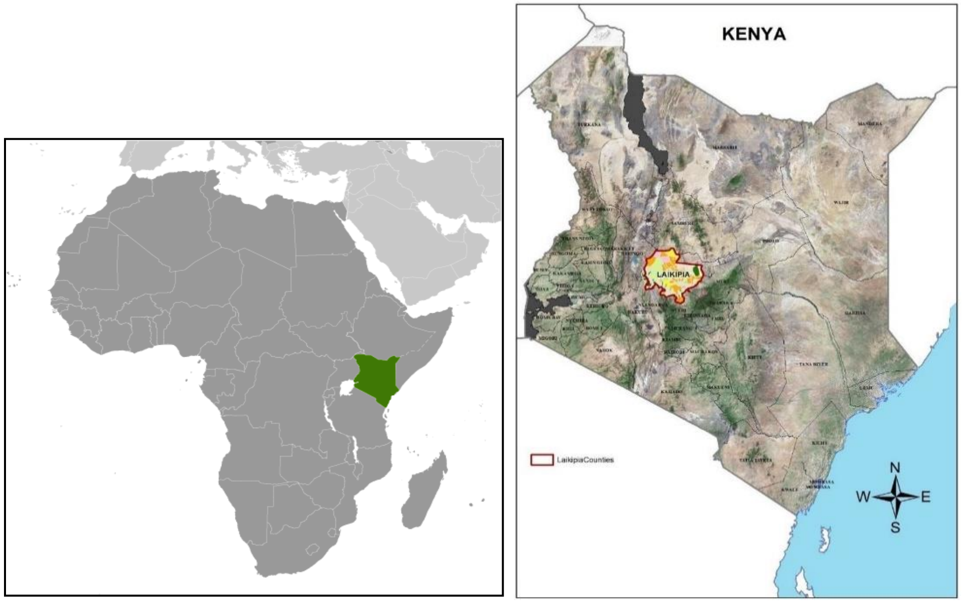

Figure 3-2. Location of Laikipia County, Kenya in East Africa.

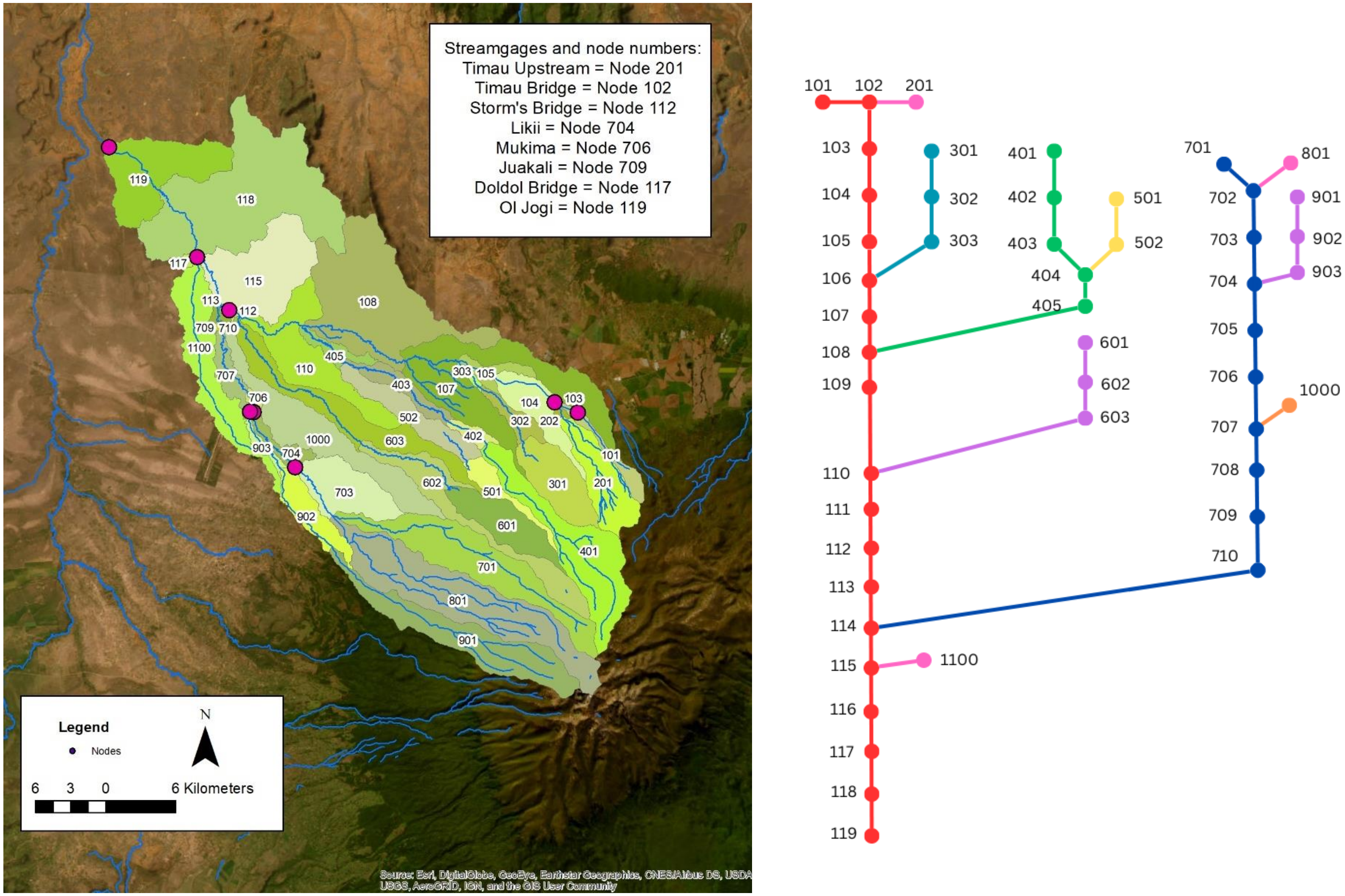

Figure 3-3. Nanyuki River watershed in Laikipia, Kenya used in the study (left) and corresponding CUENCA link-and-node diagram (right). Circles in watershed map indicate stream gauge locations, and colors in link-and-node diagram indicate different stream channels. A summary of characteristics of each node subbasin are contained in Table B-1.

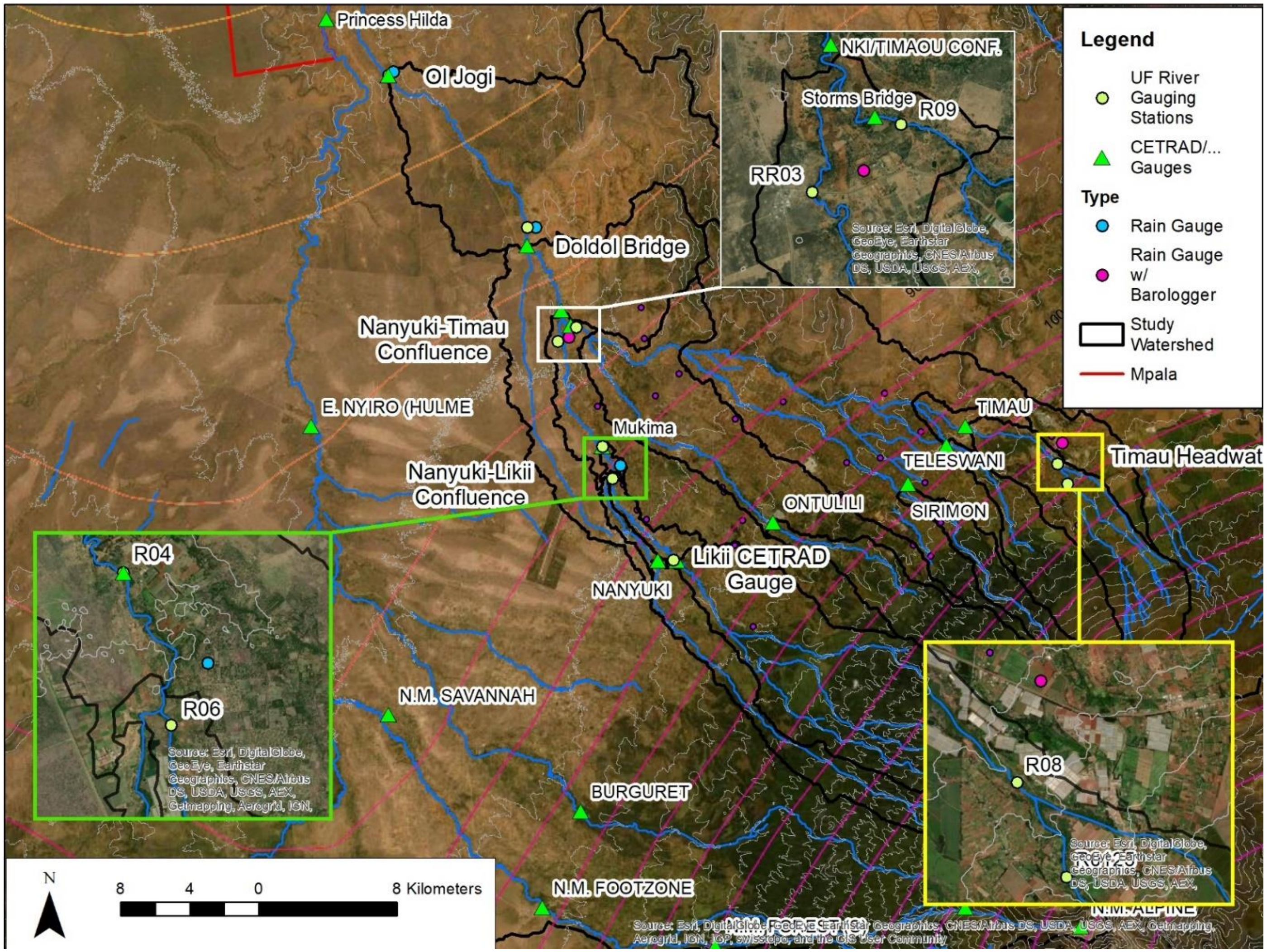

Figure 3-4. Nanyuki River watershed stream and rain gauge network. UF gauges are labeled with original names from location scouting exercise. R0125 = Timau Upstream (CA01), R08 = Timau Bridge (CA02), R09 = Storms Bridge (CA03), R06 = Likii (CA04), R04 = Mukima (CA05), RR03 = Juakali (CA06), Doldol Bridge = Doldol (CA07), Ol Jogi = Ol Jogi (CA08).

Figure 3-5. Monitoring equipment in Laikipia, Kenya. Left: Rainwise tipping bucket rain gauge with solar panel and battery box. Center: Solinst Levelogger 5 with direct read cable removed from streamgage housing below. Right: Streamgage housing visible from dry river conditions. Concrete box contains metal pipe interior and was constructed on site, with locks at top crafted by local metalworker to reduce possibility of theft. Photos courtesy of author.

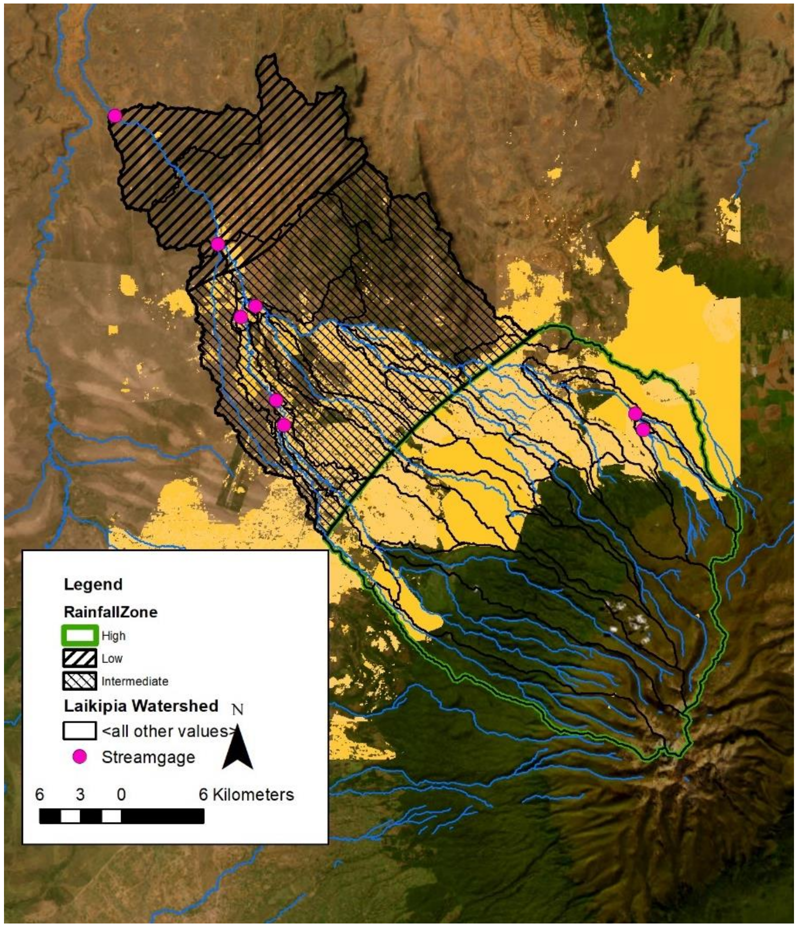

Figure 3-6. Rainfall areas and agricultural landuse within Nanyuki River Watershed. Orange/yellow areas represent agricultural area.

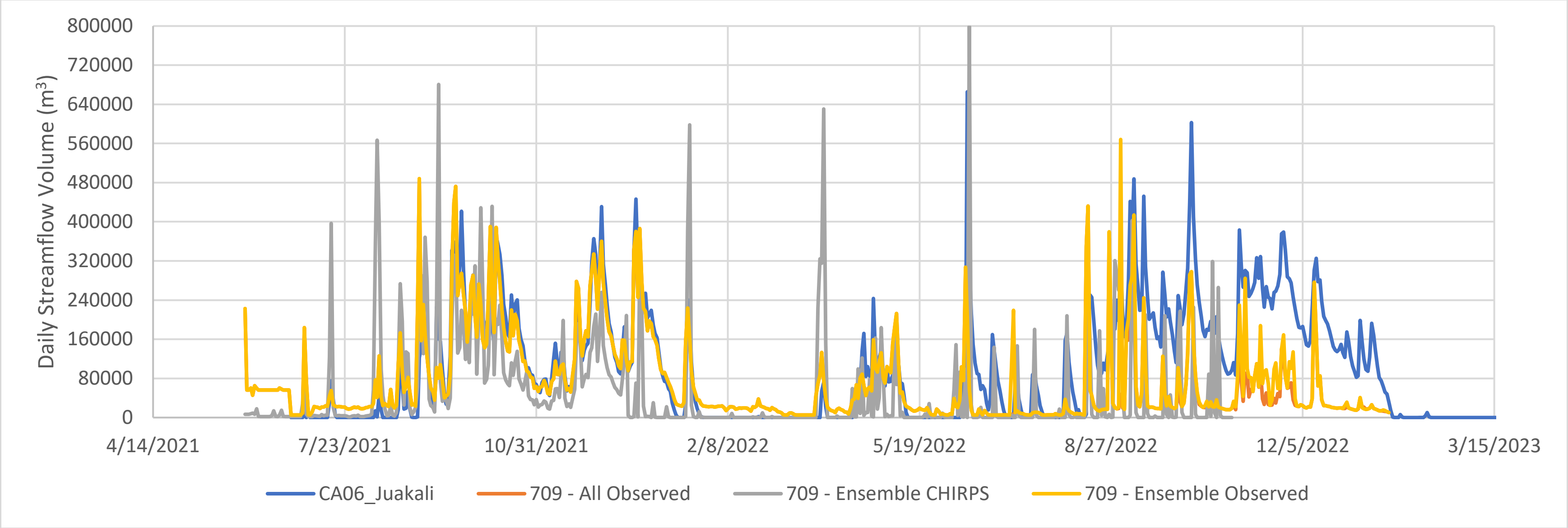

Figure 3-7. Comparison of different rainfall inputs on baseline scenario outcomes using Curve Number rainfall-runoff method. CA06 – Juakali is the observed daily streamflow. 709 is the watershed node that matches the Juakali streamgage location. "All observed" refers to a rainfall input file based entirely on observed rainfall data. "Ensemble CHIRPS" contains CHIRPS data for the subwatersheds that did not contain a rain gauge. "Ensemble Observed" includes primarily observed data with some adjustments made to add high rainfall events on Mount Kenya that were not captured with the rain gauges.

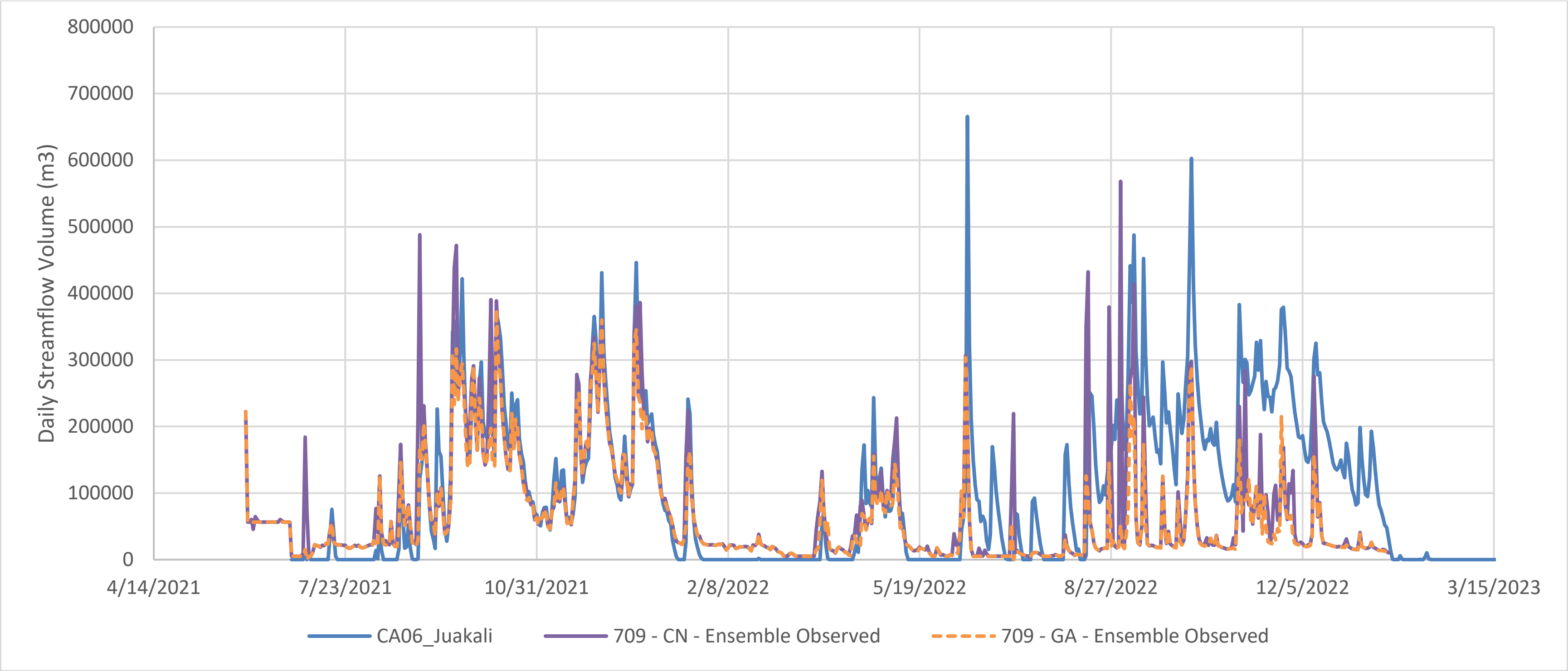

Figure 3-8. Comparison of Green-Ampt and Curve Number methods at Juakali stream gauge.

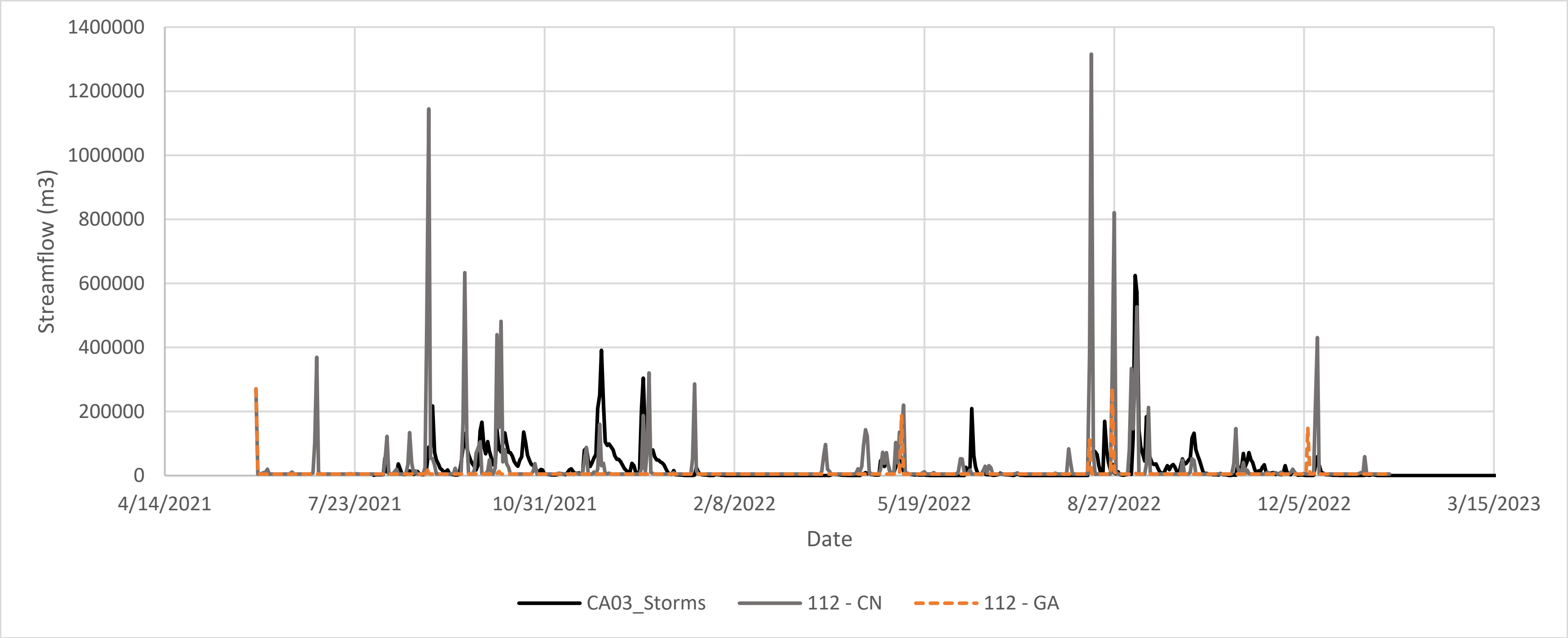

Figure 3-9. Storms bridge (Node 112) simulation results.

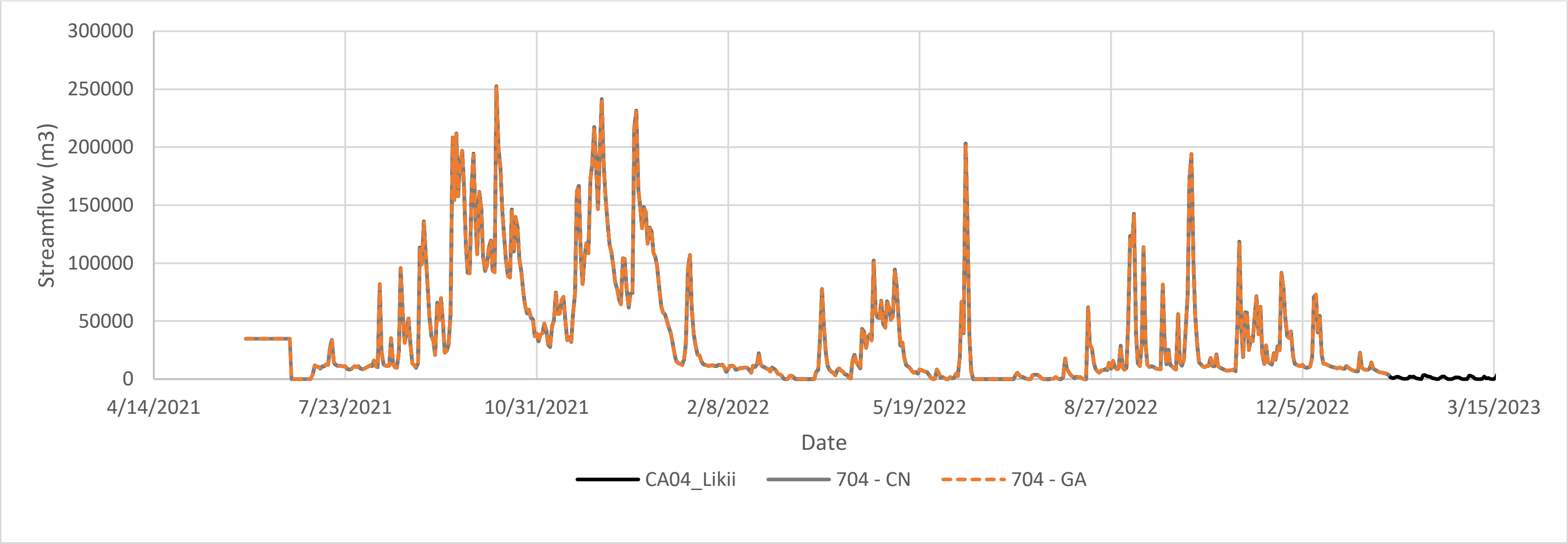

Figure 3-10. Likii River (Node 704) simulation results. This dataset was used as a model input.

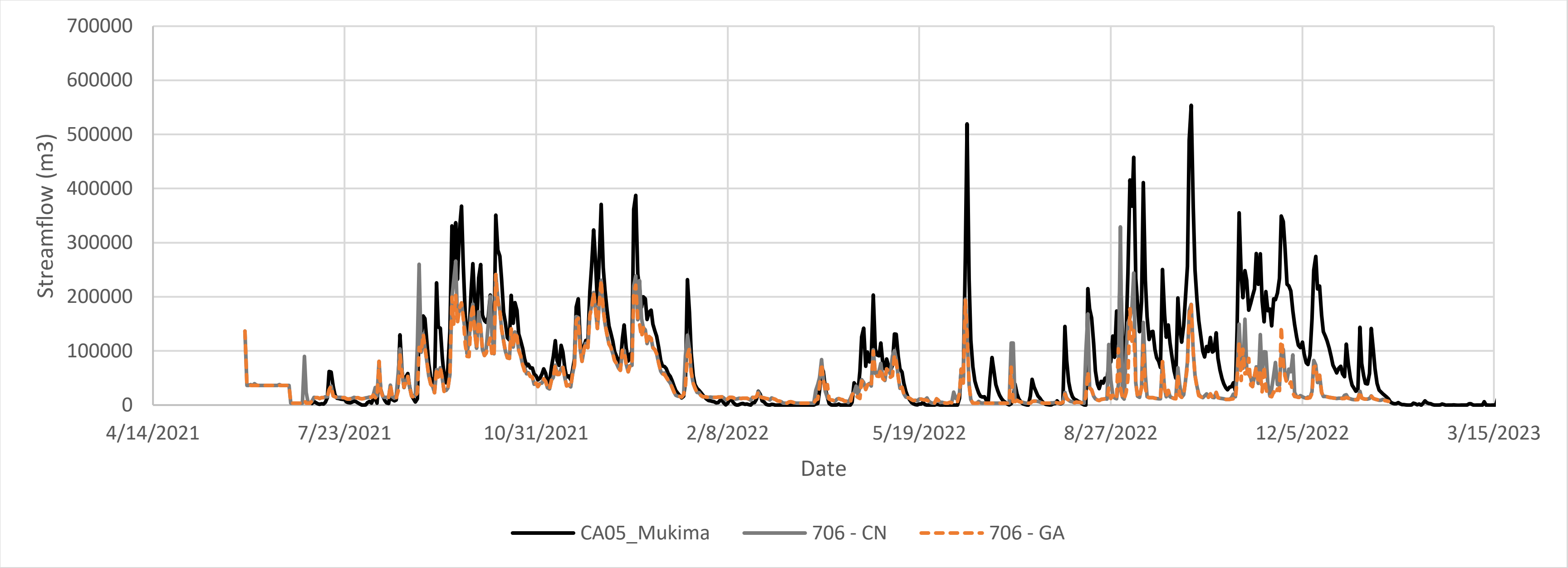

Figure 3-11. Mukima (Node 706) model results.

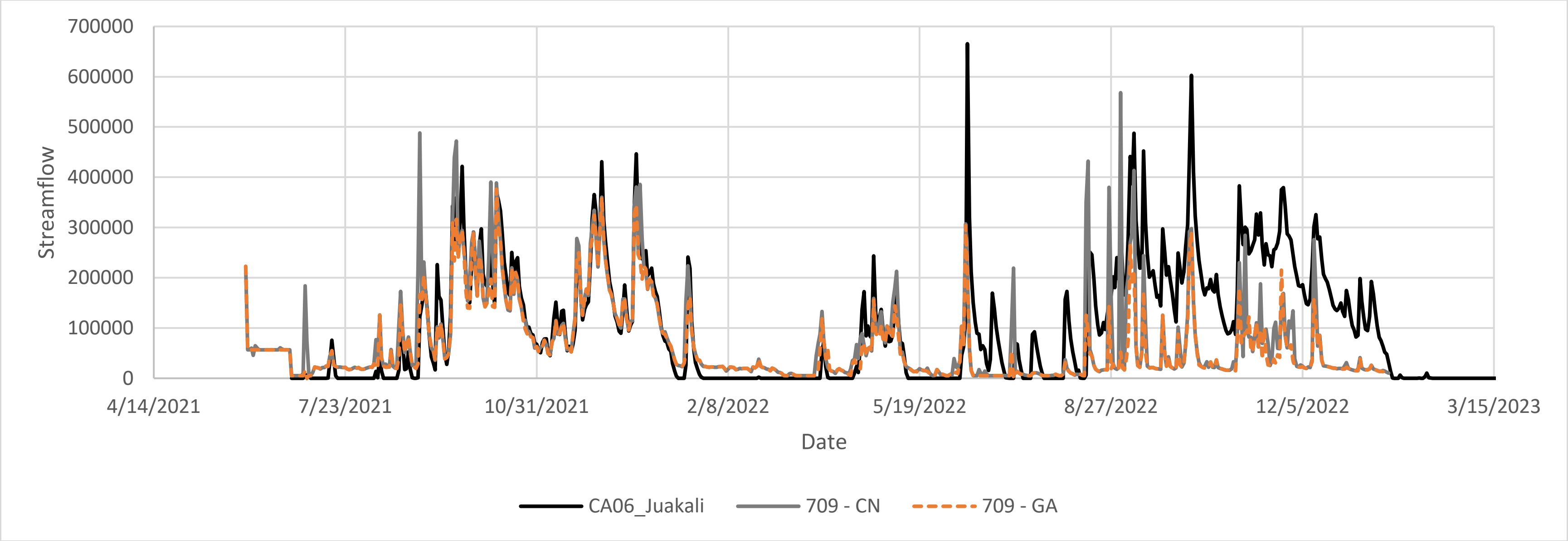

Figure 3-12. Juakali (Node 709) model results.

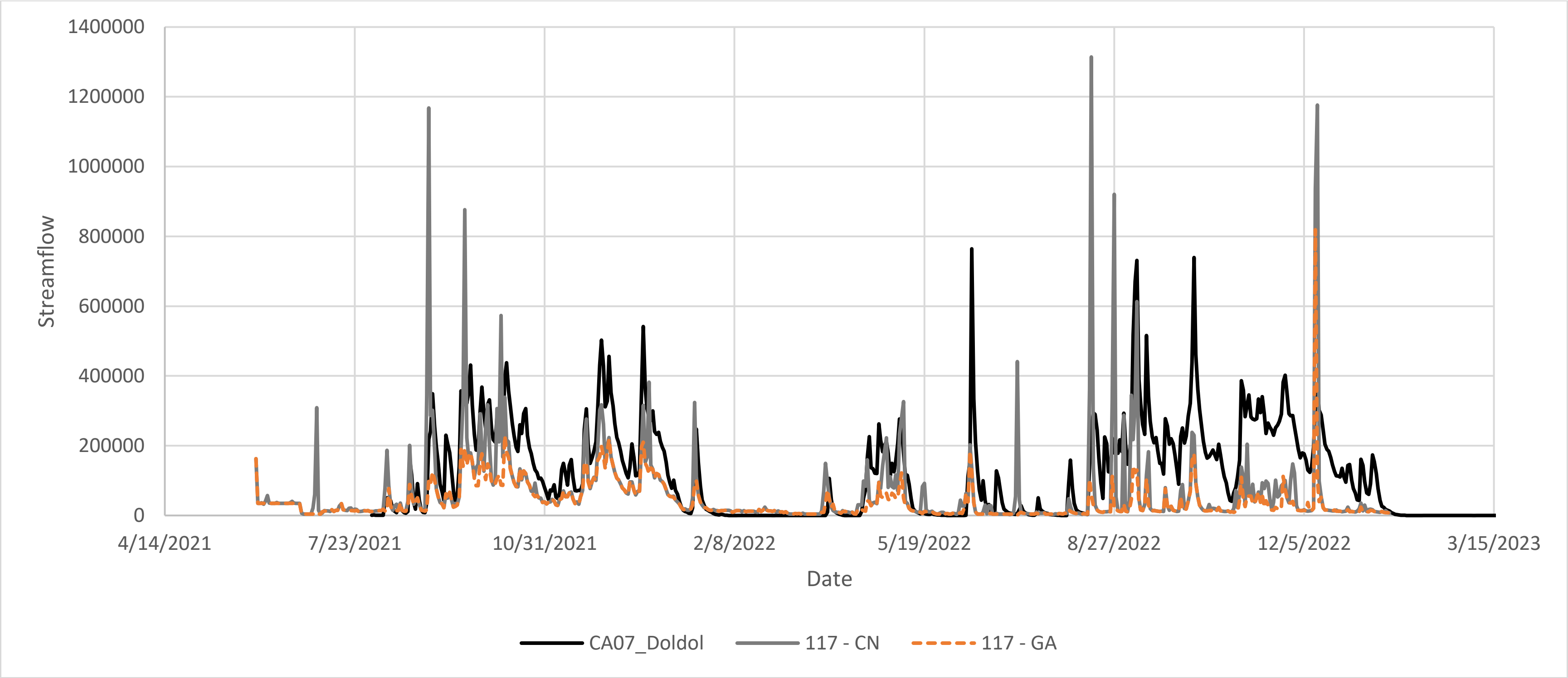

Figure 3-13. Doldol Bridge (Node 117) model results

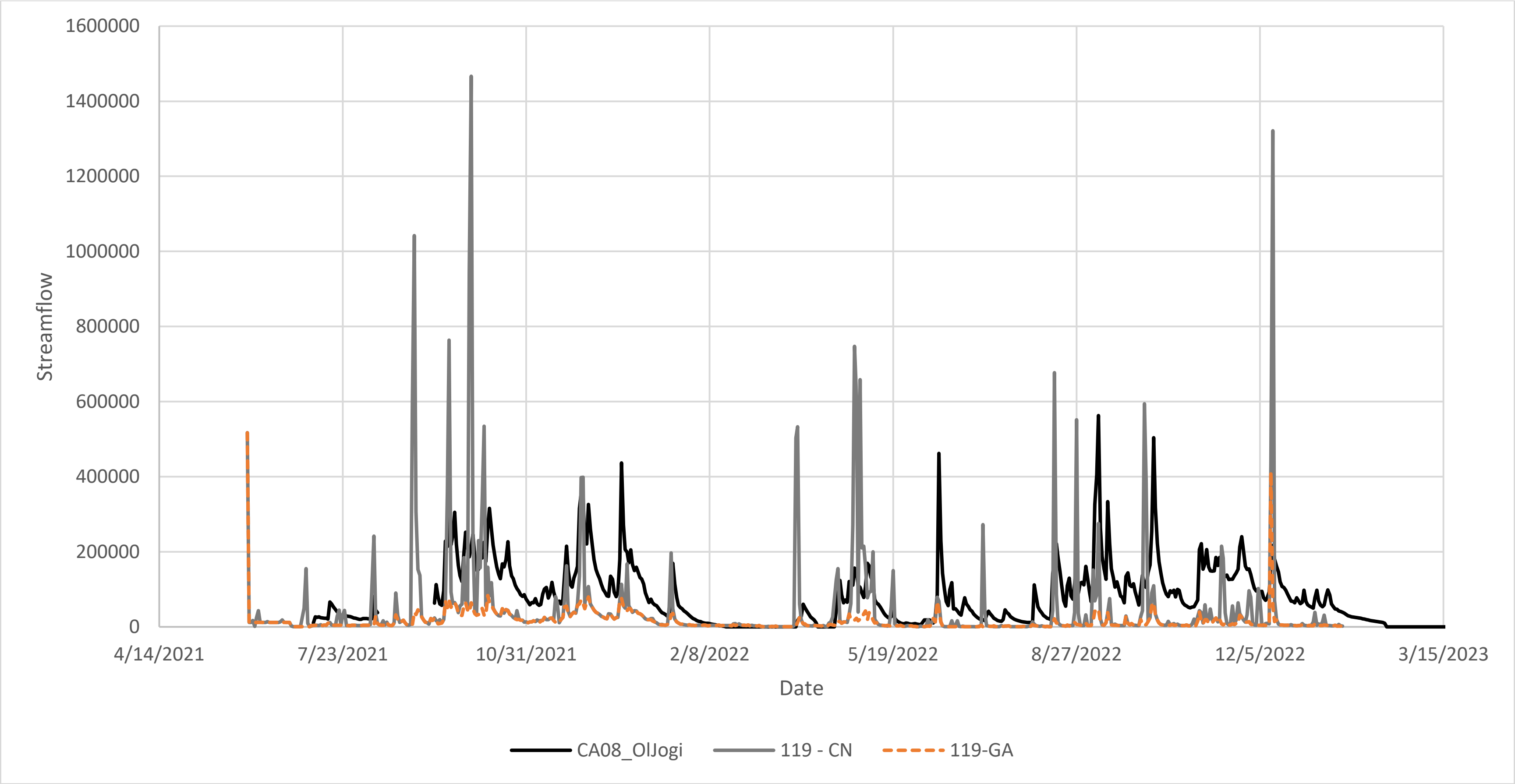

Figure 3-14. Ol Jogi (Node 119) simulation results. This is the most remote gauge located in the savanna ecosystem.

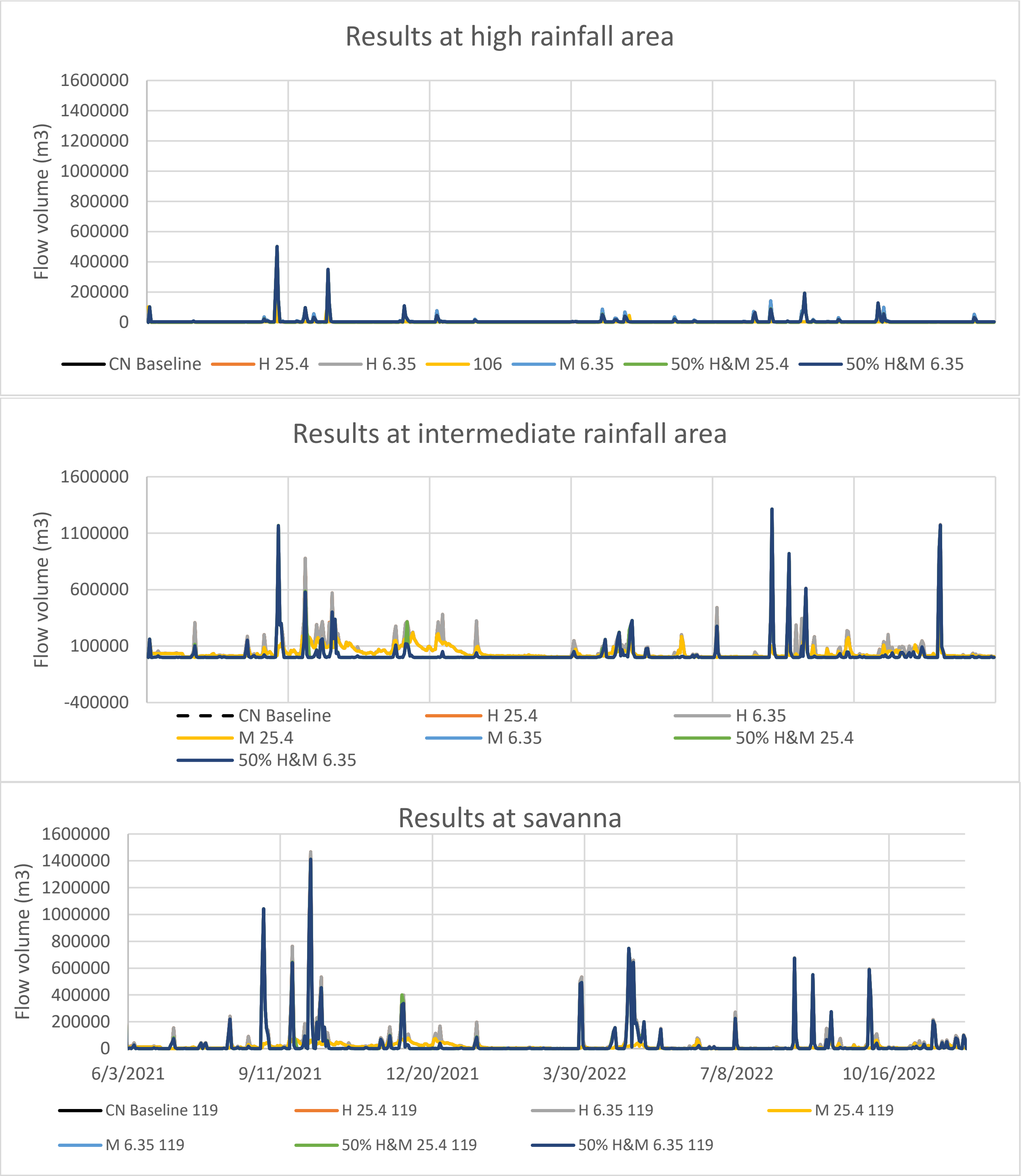

Figure 3-15. Results of irrigation scenarios (H=applied in high rainfall area, M= applied in intermediate rainfall area, 25.4 = conventional irrigation scenario, and 6.35 = drip irrigation scenario) from upstream (high rainfall area) to downstream (savanna). Graphs depict daily flow volume in cubic meters.

Table 3-1. Details of remote sensing input datasets.

Product	Source	Spatial resolution	Temporal resolution	Length of record
CHIRPS 2.0	NOAA-CPC	0.05 degrees	6-hourly, daily, monthly	1981-Present
MODIS AQUA LST	NASA LP DAAC	1000 m	Daily	2000-Present
MODIS TERRA 8-day ET	NASA LP DAAC	500 m	8-day	2001-Present
MERRA-2	NASA-GMAO	0.625 degree	hourly	1980-present

Table 3-2. Management scenarios tested in SAI analysis.

Scenario Name	Description
Baseline	No irrigation, conventional tillage
Irrigation scenarios	
H 25.4	Conventional irrigation applied in high rainfall area
H 6.35	Drip irrigation applied in high rainfall area
M 25.4	Conventional irrigation applied in intermediate rainfall area
M 6.35	Drip irrigation applied in intermediate rainfall area
H&M 25.4	Conventional irrigation applied on 50% of agricultural land throughout watershed
H&M 6.35	Drip irrigation applied on 50% of agricultural land throughout watershed
Tillage Scenarios	
NT H 100	No-till applied in 100% of high rainfall area
RT H 100	Reduced tillage applied in 100% of high rainfall area
NT M 100	No-till applied in 100% of intermediate rainfall area
RT M 100	Reduced tillage applied in 100% of intermediate rainfall area
NT H&M 100	No-till applied in 100% of agricultural land throughout watershed
RT H&M 100	Reduced tillage applied in 100% of agricultural land throughout watershed
NT H&M 50	No-till applied in 50% of agricultural land throughout watershed
RT H&M 50	Reduced tillage applied in 50% of agricultural land throughout watershed

Table 3-3. NSE and RMSE results of CN and GA models compared to baseline scenarios.

Measurement		112_Storms	704_Likii	706_Mukima	709_Juakali	117_Doldol	119_OlJogi
Entire study period	CN NSE	-2.50	1.00	0.38	0.25	-0.17	-1.83
	CN RMSE	103345	110	75512	100015	148724	138265
	GA NSE	-0.19	1.00	0.30	0.30	-0.10	-0.57
	GA RMSE	60147	110	80529	96545	139275	102791
July 2021 – June 2022	CN NSE	-2.67	1.00	0.78	0.83	0.30	-1.88
	CN RMSE	95314	133	40392	42667	103343	139612
	GA NSE	-0.25	1.00	0.76	0.92	0.38	-0.22
	GA RMSE	55652	133	42335	29201	97009	90749
June 2022 - Jan 2023	CN NSE	-2.40	1.00	0.02	-0.34	-0.79	-1.82
	CN RMSE	112076	67	104630	139003	188813	136683
	GA NSE	-0.15	1.00	-0.13	-0.46	-0.57	-1.00
	GA RMSE	65144	67	111946	144885	176667	115220

Table 3-4. Comparison of NSE values for GA Laikipia watershed simulation at Juakali streamgage for different precipitation datasets. The “all observed” dataset utilizes rain gauge data with simple spatial statistics to adjust across the Laikipia watershed. The “ensemble CHIRPS” dataset uses primarily CHIRPS data with observed data used only in the subwatersheds where it was observed. The “ensemble observed” dataset includes primarily observed data with CHIRPS data substituted when storm events were clearly missed (particularly for upper elevations of Mount Kenya) by rain gauges. The “ensemble observed” dataset performed best and was thus used in subsequent analyses.

	All Observed	Ensemble CHIRPS	Ensemble Observed
NSE	0.25	-0.24	0.30
RMSE	100,015	125977	96545

Table 3-5. Flow volume changes under different irrigation scenarios for the total study period and single-day maximum flow increases and reductions.

Irrigation Scenario	High Rainfall Area			Intermediate Rainfall Area			Savanna		
	Flow volume change, %	Maximum increase, %	Maximum reduction, %	Flow volume change, %	Maximum increase, %	Maximum reduction, %	Flow volume change, %	Maximum increase, %	Maximum reduction, %
H 25.4	-46.21	7.12	-99.61	-0.01	5.94	-2.37	0.02	2110.23	-100
H 6.35	-21.19	1.22	-42.45	-0.04	0.93	-1.50	0.00	0.29	-0.07
M 25.4	-1.73	0.00	0.00	-59.72	14.67	-100	-29.70	785.23	-100
M 6.35	0.00	0.00	0.00	-61.59	2.41	-100	-28.87	160.33	-100
H&M 25.4	-34.63	5.69	-76.89	-60.22	4.89	-100	-28.09	160.33	-100
H&M 6.35	-14.21	0.85	-42.85	-61.61	1.53	-100	-28.81	160.33	-100

Table 3-6. Flow volume changes under different tillage scenarios for the total study period and single-day maximum flow increases and reductions.

Tillage Scenario	High Rainfall Area			Intermediate Rainfall Area			Savanna		
	Flow Volume Change, %	Maximum Flow Change, %	Minimum Flow Change, %	Flow Volume Change, %	Maximum Flow Change, %	Minimum Flow Change, %	Flow Volume Change, %	Maximum Flow Change, %	Minimum Flow Change, %
NT H 100	-20.73	4.84	-54.41	-2.20	11.63	-30.60	-1.02	7.41	-31.56
NT M 100	0.00	0.00	0.00	-1.17	8.37	-16.30	-0.77	6.48	-24.50
NT H&M 100	-20.73	4.84	-54.41	-7.35	75.60	-46.04	-4.42	263.75	-72.51
NT H&M 50	-11.77	3.68	-33.12	-2.99	75.30	-43.05	-1.34	258.21	-72.51
RT H 100	-17.03	4.77	-46.20	-1.76	9.06	-24.67	-0.81	5.81	-25.35
RT M 100	0.00	0.00	0.00	-2.42	75.28	-43.05	-0.93	258.07	-25.35
RT H&M 100	-17.03	4.77	-46.20	-5.00	75.36	-43.05	-2.83	258.49	-25.35
RT H&M 50	-9.45	2.15	-27.07	-3.13	75.34	-43.05	-1.82	258.38	-25.35

Table 3-7. Comparison of CUENCA with other hydrologic models, based on assessment by Haberlandt (2010). For target variables, Q = river discharge, ET = evapotranspiration, Perc = percolation, RO = runoff, RI = interflow, RB = baseflow [RI and RB currently lumped as baseflow in CUENCA]. LULC = landuse/landcover.

Performance measurement	SWAT	HEC-HMS	WaSim-ETH	CUENCA
Typical use	LULC change, agricultural management	LULC change, agricultural management, infrastructure control	LULC change, agricultural management, infrastructure control	LULC change, agricultural management, infrastructure control
Spatial specification	Semi-distributed	Semi-distributed	Fully distributed	Semi-distributed
Temporal scale	Daily	Daily	Daily	Daily
Processes (and flexibility to choose alternative algorithms)	Best (high flexibility)	Good (high flexibility)	Good (some flexibility)	Good (little flexibility)
Target variables	Q, ET, Perc, RO, RI, RB	Q, ET, Perc, RO, RI, RB	Q, ET, Perc, RO, RI, RB	Q, ET, Perc, RO, RI, RB
Complexity (Total Parameters)	High	Intermediate	Intermediate	Intermediate
Efficiency	Good runtime Manual calibration	Fastest runtime Slowest calibration	Good runtime Intermediate calibration	Slow runtime Manual calibration
Sensitivity to a/deforestation, urbanization, crop rotations, and fertilization scenarios	Most sensitive to crop rotations	Least sensitivity to LULC and agricultural scenarios	Most sensitive to LULC change	Sensitivity testing in progress
Ease of handling	Difficult	Easiest	Intermediate	Easy

CHAPTER 4
LINKING SUSTAINABLE AGRICULTURAL INTENSIFICATION PRACTICES TO HYDROLOGIC ECOSYSTEM SERVICES IN SAVANNA LANDSCAPES

Sustainable agricultural intensification (SAI) has been conceptually touted as a method of increasing agricultural production while protecting natural resources, such as soil heath, water quantity, and water quality (Pretty et al., 2011; Rockstrom et al., 2017). While improvements to soil function have been mixed at the field scale, they have rarely been tested at the watershed scale using hydrologic modeling techniques specific and sensitive to changes in tillage (Rockstrom et al., 2009; Fan et al., 2012; Thierfelder et al., 2015; Giller et al., 2009; Mupangwa et al., 2007; Pittelkow et al., 2015). Additionally, in the US and other food-secure countries, where most research has taken place, agricultural watersheds are typically homogeneous, with large tracts of land managed homogeneously (Verhulst et al., 2010; Tomer and Locke, 2011). In food insecure regions, where smallholder farms comprise large sections of the landscape, issues of land, labor, and capital may mean that SAI, specifically with regard to reduced tillage and implementation of drip irrigation, is not feasible or that the field-scale benefits may be even more dampened by the heterogeneous nature of the landscape (Valbuena et al., 2012).

Although conversion from conventional to high efficiency irrigation is commonly cited as a water-saving measure, these savings are rarely redistributed as environmental flows (Batchelor et al., 2014). Decadal studies have determined that improved irrigation efficiency at the plot-level typically leads to reduced water availability at the watershed scale due to increased area put under agricultural cultivation or conversion to crops with high-water requirements (Grafton et al., 2018; Scott et al., 2014). Since field studies are difficult to implement under a controlled environment, modeling studies are often used to evaluate different irrigation scenarios. Even these studies can be difficult to compare due to varying definitions in efficiency and inconsistent methods in ET dynamics that may not accurately reflect reality (van der Kooij et al., 2013). Basin-level evaluations geared towards environmental flows have indicated that high efficiency water use can preserve environmental flows based on the ratio of precipitation to evapotranspiration (Batchelor et al., 2014). Similarly, negative impacts of high efficiency irrigation, such as reduction in groundwater recharge, are negligible compared to conventional irrigation during wet years (Pool et al., 2022). However, landscape water storage (such as rainwater harvesting, small detention ponds, etc) with high efficiency irrigation may be more likely to both increase crop yields and reduce negative impacts on downstream flows (Baker et al., 2012).

Since reduced and no-till practices have variable impacts at the field level, watershed level impacts are also varied. Saturated hydraulic conductivity (Ksat) impacts are highly context specific, dependent on local rainfall intensity, ET, and soil texture, as well as time under conservation tillage management (Strudley et al., 2008; Tomer and Locke, 2011; Bosch et al., 2012; Didone et al., 2014; Endale et al., 2014; Easwaran et al., 2021). It is also difficult to attribute any changes in landscape hydraulic properties to overall watershed hydrology (Bowmer, 2011). Studies have shown evidence of increased baseflow recovery (i.e. low flow conditions recover to “normal” faster) and reduced irrigation requirements for similar crop yields, both indicators of improved soil water holding capacity and soil structure (Tomer et al., 2005; Baumhardt et al., 2017; Assefa et al., 2018). Studies have also shown peak storm flow reduction (especially with cover crops) in conservation agriculture systems (Andraski et al., 1985; Yog and Rochester, 1989)

Flow duration curves have been used widely since the 1950s to characterize regional flow patterns in the United States (Vogel and Fennessey, 1994). They are extensively used in hydrologic studies of water quality, watershed management, flood assessment and mitigation, drought assessment, groundwater recharge, landuse and landcover change, soil conservation, environmental flows, and climate change (Leong and Yokoo, 2021). The shape of the curve can give insights into many characteristics affecting flow, including precipitation, landuse, and geology (Leong and Yokoo, 2021). For example, the steepness of the curve indicates the catchment's ability to store or transfer precipitation, and therefore can also be used as a model evaluation tool to estimate whether a process-based model is properly simulating landscape storage (Yilmaz et al., 2008). In another example, significant human alterations, such as the installation of a dam or flow control structure, will reduce the variability of natural flows and create flat sections and/or vertical sections within the curve (Basso and Botter, 2012).

In Laikipia, Kenya, stakeholders throughout the watershed could benefit from upstream agricultural management changes. Along the Nanyuki River, a tributary of the Ewaso N'giro, the steep rainfall gradient provides upstream farmers with significantly more access to both river water and rainfall as sources of irrigation and household water. Improved resource use efficiency in high rainfall areas could benefit mid-watershed stakeholders, who are typically farmers and pastoralists in the dryer region of Laikipia, County. At the farthest reaches, it could benefit far savanna ecosystems and agricultural communities, including ranches and wildlife conservancies.

The Grevy's zebra has been listed as endangered by the ICUN, with a population decline from 15,000 to about 2,000 from the 1970s to today. Its range today only includes Northern Kenya, including Laikipia County, and parts of Ethiopia. Increased prevalence of drought has the potential to significantly affect population levels due to death of foals. Grevy's zebra habitat suitability is highly linked to proximity to water (Smith et al., 2022). Grevy's zebra foals must drink water daily, meaning that mares must stay close to a consistent water supply. Adult Grevy's are slightly more drought tolerant, requiring water every three days (Churcher, 1993). In areas of the savanna that get little rainfall, once watering holes have dried, this means the zebras must stay near a river as a source of water. Although foaling times are typically from April to June, when long rains are present in Laikipia, mares can give birth all year round (Becker and Ginsberg 1990). With increasingly erratic rainy seasons in Kenya, it may become more difficult for mares and foals to reach water, meaning death of offspring and declining populations. In addition, Grevy's tend to prefer habitat with low cattle density, which means that as competition for water resources increases, Grevy's will have less suitable habitat to support their population (Smith et al., 2022).

We hypothesize that by implementing substantial SAI practices in the Nanyuki River watershed, the duration of days without water flow in the downstream savanna region will be reduced when compared to existing conditions, however no flow conditions will persist in the dry season.

4.1 Methods

Study region. Laikipia, Kenya is a county in the Mount Kenya region of Kenya that is experiencing rapid population and agricultural growth at the edge of a savanna ecosystem that supports important wildlife. Since the 1980s, Laikipia has been experiencing agricultural expansion, first as extensification where new lands were converted to agriculture, and later (after 2000) as intensification when existing agricultural lands increased production through irrigation (Eckert et al., 2017). Laikipia has similarly had up to 20% of its land area convert to urban area in the same time period (Muriithi, 2016). And, during a similar time period, the arid areas of northern Kenya have lost on average 68% of their wildlife while increasing in livestock biomass (Ogutu et al., 2016). To maintain the remaining wildlife, which includes the endangered Grevy's zebra, water management strategies must be implemented to attempt to buffer against climate change impacts and low flows exacerbated by human consumption (Ogutu et al., 2016).

Scenarios were developed to test impacts of SAI practices on hydrology in the Nanyuki watershed. These scenarios, described in Chapter 3, are based on the hypotheses described in Chapter 1. Table 4-1 describes the scenarios tested. The uncalibrated CUENCA link-and-node hydrologic model was used to simulate each scenario using the Curve Number rainfall-runoff method.

Flow duration curves were developed for each scenario by calculating the exceedance probability of each daily flow value during the simulation period (Searcy, 1959). Figure 4-1 outlines the methods used in this chapter. For each scenario and for observed flows at Ol Jogi, flow values during the simulation period were ranked from highest to lowest. Then, the exceedance probability was calculated where:

$$P = 100 * (M/(n+1)) \qquad \text{(Eq 1)}$$

Where P = exceedance probability (%), M = the ranked position, and n = the total number of events during the time period of interest.

The Grevy’s zebra is sensitive to even one day without flow if it is during foaling season. Realistically, in the Nanyuki and Ewaso N’giro Rivers, it takes several days of no-flow before the stream channel dries out completely, because deep pools in the channel and in the landscape will retain water. Unfortunately, flow-duration curves do not specify the exact timing of flow occurrences, and therefore whether low-flow days are consecutive (Leong and Yokoo, 2021). Therefore, any shift to the right of the flow duration curve is considered an improvement to the baseline scenario (Smakhtin and Eriyagama, 2008), but a shift of 3% exceedance probability or more (11 days fewer dry days per year) is considered a significant achievement to reduce risk to the Grevy’s zebra during these scenarios. In addition, daily no-flow values were typically observed in Ol Jogi once flow was below 1200 m^3 daily, so this threshold is used as a cutoff point to evaluate lateral shifts in the flow duration curve.

4.2 Results

Figure 4-3 shows the model results for irrigation scenarios compared to observed stream flow at Ol Jogi streamgage in the savanna region of the Nanyuki River watershed. During the data collection period from August 2021 to July 2023, two periods of no-flow conditions were observed, with one lasting 40 days and the other lasting 12 days. Neither the GA or CN model predicted this, but there were several periods of consistent low flow at approximately 630 m^3 per day (visible in Figure 4-2). Improvements to the model, such as using soil property values from field observations, could change these results. For the irrigation specific scenarios, drip and conventional irrigation in the high rainfall area (H 6.35 and H 25.4) predict the same model outcomes as the curve number algorithm. These results are shifted about 1% to the right of the GA baseline, conventional irrigation in intermediate rainfall areas, and the Ol Jogi observed data at the 1200 m^3 threshold, with an exceedance probability of 88% compared to 89% for the former group. All the other scenarios (H&M 25.4 at 50%, M 6.35, and H&M 6.35) are significantly drier, with exceedance probabilities for 1200 m^3 daily between 33 and 35%.

The results of the no-flow frequency for the tillage scenarios are shown in Figure 4-5. None of the scenarios predicted zero flow days, but they contained multiple consecutive days with flow at 632 m^3 which seemed to be a minimum flow value. At the low-flow value of 1200 m^3, all models performed essentially the same with a 89-90% exceedance probability, similar to Ol Jogi.

Across the entire flow duration curve for both irrigation and tillage, flow dynamics did not quite match those observed at Ol Jogi. Under the irrigation scenarios, the baseline irrigation and intermediate conventional irrigation had abrupt shifts from low flows to high flows. The other scenarios had more flow variability, but the graph is shifted downward from the Ol Jogi observed data, indicating higher probability of overall lower flows, which is what we observe in Figure 4-2. From a model evaluation perspective, this indicates that rainfall may not be stored properly in the landscape (i.e. soil water storage or ponding) to simulate flow dynamics. The tillage scenarios all look similar to the second group of irrigation scenarios, so there is most likely the same underlying process error in the model.

4.3 Discussion

The irrigation scenario results reflect observations of the irrigation efficiency paradox (Grafton, 2018), in which a transition to high-efficiency irrigation increases water losses to evapotranspiration and decreases losses to surface runoff. During the drip irrigation scenarios, irrigation water primarily contributes to plant processes or minor increases in soil moisture, and almost none returns to the stream channel as return flows unless the soil is already saturated. Since these scenarios use streamflow as the only source of irrigation, these results are not surprising. More advanced model features, such as incorporation of boreholes, irrigation ponds, or rainwater harvesting strategies, would alter these results. These results are consistent with several studies that indicate that water savings from drip irrigation either do not scale up (Pool et al., 2021; Scott et al., 2013) or that the water savings were not considered holistically within basin water to begin with (van der Kooij et al., 2013).

The tillage results are largely similar among the scenarios, and with the poor prediction capacity of the model, the differences are most likely insignificant. A thorough look at no-flow days in the model results indicates that the no-till scenarios implemented throughout the watershed have the highest instances of single low flow days, and they have slightly different distributions for consecutive low flow days, with 100% no-till coverage having one less instance of 5 consecutive low flow days (2 instead of 3), and 50% no-till having one less instance of 2 consecutive low flow days. This could indicate that the NT scenarios do contribute to baseflow resilience, as others have observed (Tomer et al., 2005; Baumhardt et al., 2017; Assefa et al., 2018). As a result, small increases in baseflow do affect flow timing enough to maintain some minimum flows, and if the river does dry, Grevy's zebras would be able to access water with less travel towards urban areas. However, none of the tillage scenarios significantly impact the Grevy's risk, while adoption of irrigation in high rainfall areas rather than low rainfall does.

Both of these scenarios are important to analyze further using the Green-Ampt method, which was shown to be more sensitive to initial soil water content in Chapter 2. As mentioned in Chapter 3, the baseline model should be improved to reflect field conditions (i.e. existing saturated hydraulic conductivity, presence of volcanic soils, and long-term changes in physical properties due to no-till or reduced till). Long-term impacts of no-till are highly site-specific, and as new data becomes available on long-term field trials in Sub-Saharan Africa, results can be incorporated into this work to parameterize the tillage scenarios accurately.

4.4 Conclusions

The results of this study indicate that targeting SAI practice implementation in a heterogenous watershed such as the Nanyuki River watershed can benefit comparatively remote ecosystems. Utilizing drip irrigation in regions that receive more rainfall provides the benefits of high efficiency irrigation without significant impacts to streamflow volume in the far savanna, and is significantly better for avoiding the risk of low flows than adoption in intermediate areas. Adoption of conservation tillage is a less specific on a location, but some practices spread over a large area may be more beneficial than high density practices in a small area. Since the baseline CN model contained no zero-flow days, these particular scenarios could only be compared against one another, and drip irrigation implemented in high rainfall areas is the only scenario that performed as well as the baseline scenario. However, this study is a starting point to quantitatively evaluate SAI practices and their impacts on surrounding ecosystems at a daily timescale. In Laikipia, modeling studies have primarily used SWAT (Ngigi et al., 2007) or simpler water balance means (Ngigi et al., 2006) with a focus on the local water balance at a monthly or cropping-season timescale.

Significant study limitations should be addressed. Due to limited data, it was difficult to accurately reflect irrigation practices, and specifically water sources, in the region. Additionally, CUENCA does not take groundwater abstraction into account, which could be a significant source of irrigation water in the future. Green-Ampt parameter development is limited to the USDA soil textural classes, and data collected by the author should be assessed to accurately reflect volcanic soil conditions in the region.

SAI implementation within the Nanyuki River watershed will impact water resources throughout the watershed. While it is easier to connect changes in hydrology at the local scale, it is important to assess how management actions affect connected ecosystems for endangered species such as the Grevy’s zebra. Targeted SAI implementation is necessary to see any watershed scale benefits to ecosystems services. This is important as counties and countries continue to make recommendations for agricultural management strategies in order to protect species. Future work should continue to refine ideal locations for SAI, particularly in highly heterogeneous watersheds. Additionally, soil responses to conservation tillage under smallholder practices and large farms should be recognized and addressed in these studies, since often times smallholders are unable to maintain conservation practices longer than two years. In addition, advanced model calibration work , including Sobol global sensitivity and uncertainty analysis, will be utilized to understand influential factors in the watershed model, and thus refine the results of the flow duration curve. An uncertainty analysis will quantify the bounds of uncertainty on the flow duration curves, as well.

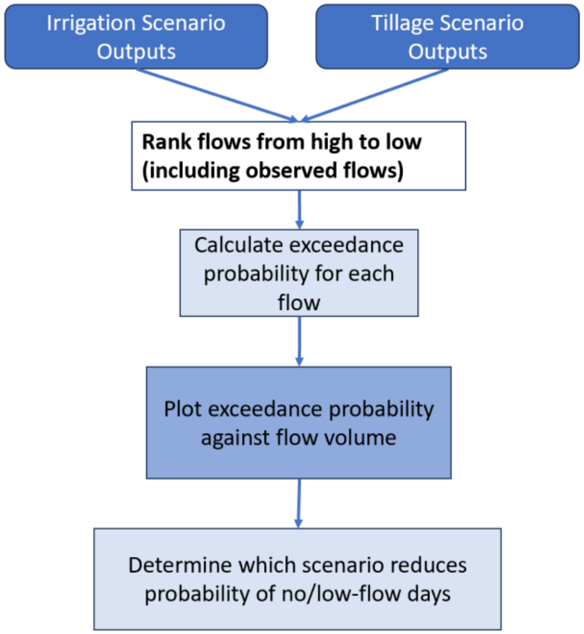

Figure 4-1. Methods for evaluating tillage and irrigation scenarios for reduction of no/low-flow days to reduce risk to Grevy’s zebra.

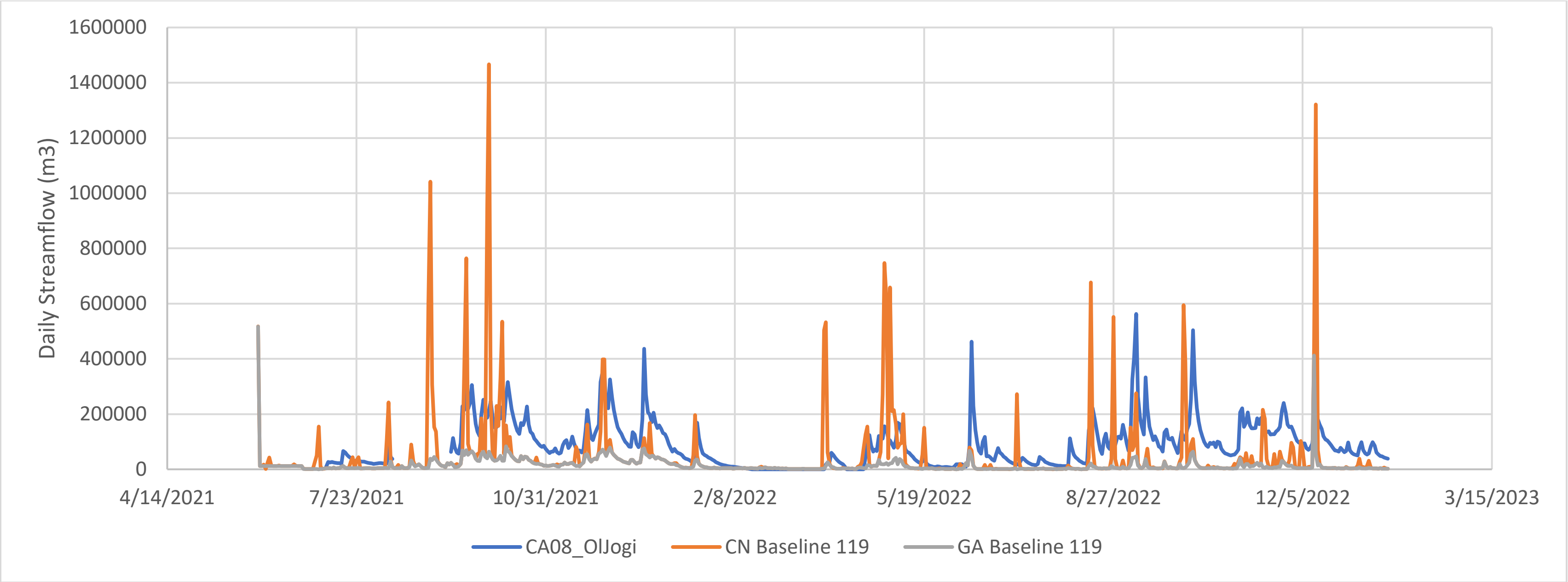

Figure 4-2. Observed streamflow (CA08_OlJogi) plotted with baseline Laikipia scenario with no irrigation simulated rainfall-runoff processes using either curve number (CN) or Green-Ampt (GA) method.

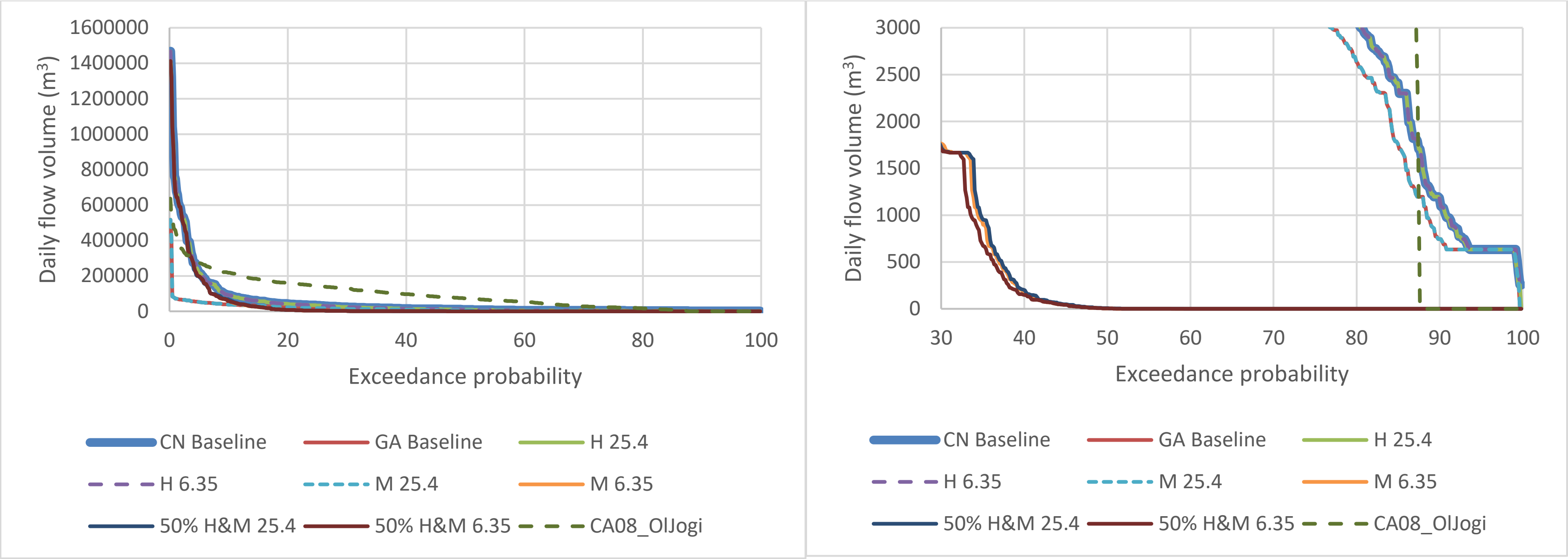

Figure 4-3. Flow duration curves for irrigation scenarios. The left figure shows the entire curve, while the right is focused on low flow dynamics. Irrigation scenarios H25.4, H6.35, and the CN baseline have the highest exceedance probabilities for low flows, indicating that there are the fewest instances of flow stopping completely.

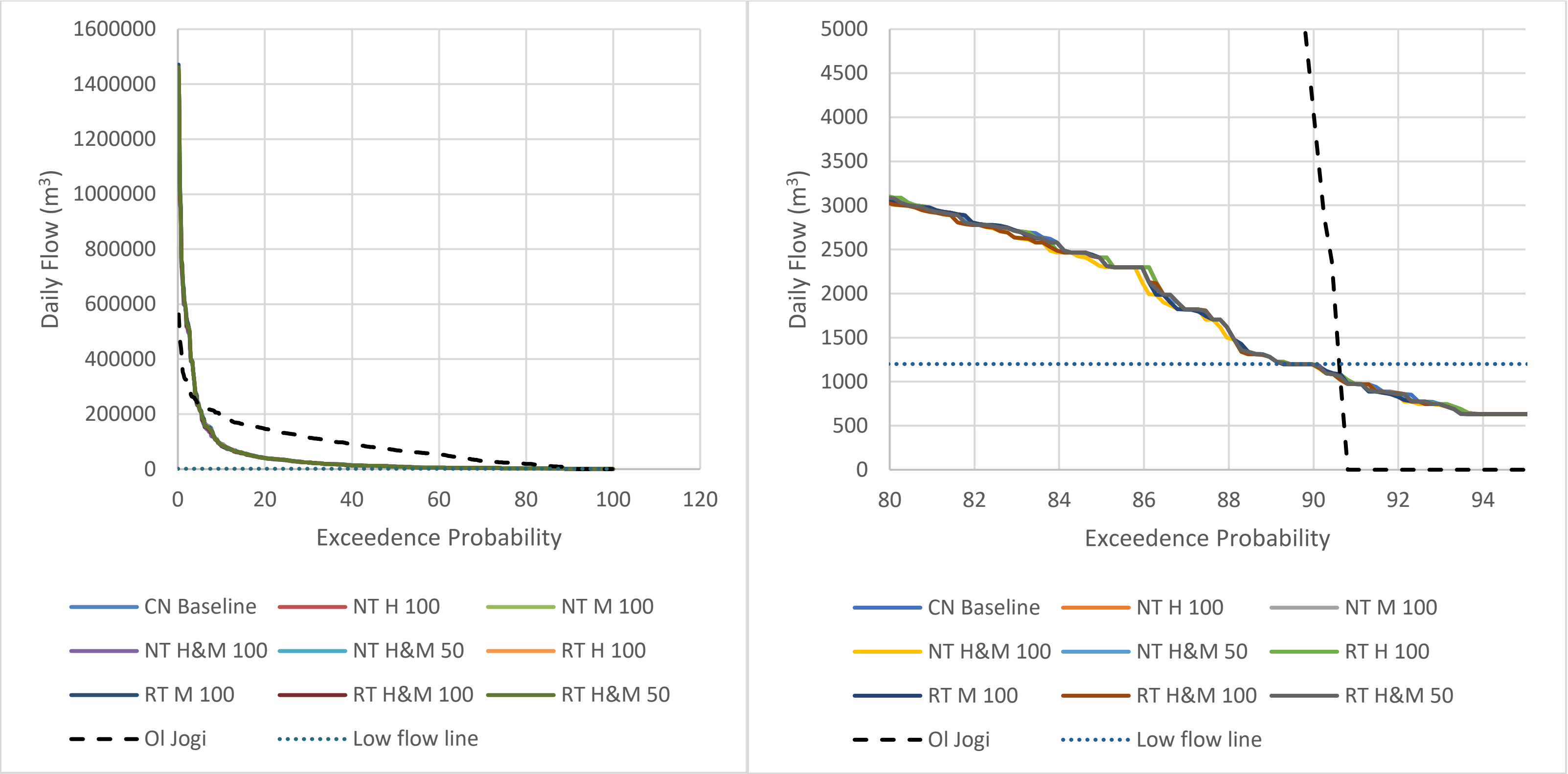

Figure 4-4. Flow duration curves under different tillage scenarios. Scenarios include no-till (NT) and reduced tillage (RT) implemented in high (H), intermediate (M), and both high and intermediate (H&M) rainfall areas at 50% and 100% coverage. The left figure shows the entire curve and the right is focused on low-flow values, where all of the scenarios perform roughly the same.

Table 4-1. Management scenarios tested in SAI analysis.

Scenario Name	Description
Baseline	No irrigation, conventional tillage
Irrigation scenarios	
H 25.4	Conventional irrigation applied in high rainfall area
H 6.35	Drip irrigation applied in high rainfall area
M 25.4	Conventional irrigation applied in intermediate rainfall area
M 6.35	Drip irrigation applied in intermediate rainfall area
H&M 25.4	Conventional irrigation applied on 50% of agricultural land throughout watershed
H&M 6.35	Drip irrigation applied on 50% of agricultural land throughout watershed
Tillage Scenarios	
NT H 100	No-till applied in 100% of high rainfall area
RT H 100	Reduced tillage applied in 100% of high rainfall area
NT M 100	No-till applied in 100% of intermediate rainfall area
RT M 100	Reduced tillage applied in 100% of intermediate rainfall area
NT H&M 100	No-till applied in 100% of agricultural land throughout watershed
RT H&M 100	Reduced tillage applied in 100% of agricultural land throughout watershed
NT H&M 50	No-till applied in 50% of agricultural land throughout watershed
RT H&M 50	Reduced tillage applied in 50% of agricultural land throughout watershed

CHAPTER 5
GENERAL CONCLUSIONS

This study has contributed a first step in connecting impacts of SAI to the broader landscape, and specifically ecosystem services connected to watershed function, in Laikipia Kenya. In Chapter 2, we developed a simple process-based model that can be used to analyze impacts of landuse change and management practices that impact soil physical health. This model is in the early stages of development, and future research should work on improving the predictive ability in the Nanyuki River watershed. This should first be attempted through improved data inputs, including potentially scaling rainfall to account for precipitation on Mount Kenya, verifying soil physical property data using field measurements, considering the hydrologic properties of volcanic soils, using an advanced calibration technique, and evaluating uncertainty associated with streamflow values. Process improvements can be made by updating the antecedent moisture condition for CN to reflect ongoing cumulative rainfall or soil moisture, as recent studies have suggested.

CUENCA includes a GA infiltration process to simulate changes in tillage, such as a conversion to reduced tillage. Therefore, in Chapter 3 the uncalibrated model was used to assess different levels of SAI adoption throughout the Laikipia watershed. Through scenario analysis, it seems that drip irrigation utilized in the high rainfall area has the smallest negative impact on streamflow downstream. Conversion to no-till did not have as clear of an impact based on where it was located, but instead seemed beneficial to have large coverage of conservation tillage methods across the watershed to reduce peak flows and improve baseflow. To improve on this work, the GA method should also be used to evaluate and compare scenarios. Because it depends on soil moisture, it might provide more dynamic responses to rainfall than the CN method.

In Chapter 4, the SAI scenarios were extended to evaluate their teleconnections to savanna ecosystems by analyzing consecutive dry-day distributions to characterize potentially risk to the Grevy's zebra. The drip irrigation scenario in high rainfall areas performed best, having no day of no-flow in the river. Conventional irrigation in high rainfall areas outperformed drip irrigation in intermediate rainfall areas. Once again, all of the tillage scenarios were similar. These results indicate that drip irrigation can negatively affect the water balance, and future research should refine the scenarios as well as the irrigation methods in the model to make sure current conditions in Laikipia are accurately reflected. There are currently many different types of irrigation in the region, including using harvesting rainwater, furrow irrigation, sprinkler irrigation, boreholes, and some drip irrigation. Ol Jogi's observed data showed that the stream dried during the study period for a little more than a month, so identifying current irrigation conditions is necessary to continue this work with the local context. Ultimately, it seems that improved irrigation practices are most suited to higher rainfall areas, while conservation tillage is beneficial across the entire watershed. All of the scenarios contributed to overall reduced flow volumes, indicating that there is considerable work that must be done to select optimal practices for smallholder and large farmers in the region.

This study attempts to assess SAI scenarios using a relatively simple process based model that could be utilized in data scarce regions in the future. As the commonly used models in the United States either continue to expand (Gassman et al., 2010) or have extensive calibration requirements (Dariane et al., 2016), CUENCA can fill a role for a simple model that is sensitive to processes of interest and not overly parameterized or auto-calibrated. With future work, including global sensitivity and uncertainty analysis, and some calibration methods to explore process dynamics, model performance in Laikipia should improve. Existing watershed models are continuously used to evaluate agricultural and water management scenarios, providing managers with monthly changes in metrics such as water yield and water quality. However, to assess impacts of landuse change and landscape management decisions on streamflow, an accurate daily response is needed (Vigerstol and Aukema, 2011). Although CUENCA does have room to improve streamflow simulations, it does reflect dynamics in the Laikipia watershed. This work indicates that sustainable intensification practices, when practiced on existing agricultural land in a watershed that contains a mixture of urban, agricultural, and savanna areas, do not scale up to noticeable improvements at the watershed level to maintain environmental flows for species of concern, such as the Grevy’s zebra, for a significant difference to flow levels during dry periods.

APPENDIX A
WATER BALANCE OF 30-DAY CLAY SOIL SIMULATION

Table A-1. Water balance components of clay sensitivity test.

DAY	Node	Process	Precipitation	Runoff	Evapotranspiration	Seepage	BaseF (TF)	Baseflow	Groundwater Recharge	Flows in excess of 24 hours	Detention	Infiltration	Root zone change in soil moisture	Subsurface change in soil moisture	Total error	Percent Error
1	101	11	25400	6551	0	0	65070	0	0	205.7	0	18640	18640	-65070	0	0
2	101	11	0	263.1	4058	0	0	0	0	-263.1	0	0	-4058	0	0	0
3	101	11	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
4	101	11	12000	124	0	0	0	0	0	24.55	0	11850	11850	0	0	0
5	101	11	0	25.58	4058	0	0	0	0	-25.58	0	0	-4058	0	0	0
6	101	11	22000	4960	0	0	0	0	0	153.3	0	16890	11680	5206	0	0
7	101	11	9000	210.2	0	0	0	0	0	-190	0	8980	0	8980	0	0
8	101	11	0	6.724	4058	0	0	0	0	-6.724	0	0	-4058	0	0	0
9	101	11	10000	839.4	0	0	0	0	0	38.67	0	9122	4058	5064	0	0
10	101	11	0	45.86	4058	0	0	0	0	-45.86	0	0	-4058	0	0	0
11	101	11	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
12	101	11	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
13	101	11	1000	0	4058	0	0	0	0	0	0	1000	-3058	0	0	0
14	101	11	20000	918.8	0	0	0	0	0	104.4	0	18980	15230	3743	0	0
15	101	11	3000	112.4	4058	0	54550	0	0	-112.4	0	3000	-1058	-54550	0	0
16	101	11	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
17	101	11	15000	339.2	0	0	0	0	0	50.19	0	14610	5117	9494	0	0
18	101	11	0	53.1	4058	0	0	0	0	-53.1	0	0	-4058	0	0	0
19	101	11	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
20	101	11	15000	339.2	0	0	0	0	0	50.19	0	14610	8117	6494	0	0
21	101	11	7000	53.1	0	0	0	0	0	-53.1	0	7000	0	7000	0	0
22	101	11	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
23	101	11	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
24	101	11	4000	0	0	0	0	0	0	0	0	4000	4000	0	0	0
25	101	11	13000	184.7	0	0	0	0	0	31.93	0	12780	4117	8667	0	0

Table A-1. Continued

DAY	Node	Process	Precipitation	Runoff	Evapotranspiration	Seepage	BaseF (TF)	Baseflow	Groundwater Recharge	Flows in excess of 24 hours	Detention	Infiltration	Root zone change in soil moisture	Subsurface change in soil moisture	Total error	Percent Error
26	101	11	15000	372.7	0	0	0	0	0	16.71	0	14610	0	14610	0	0
27	101	11	0	53.1	4058	0	0	0	0	-53.1	0	0	-4058	0	0	0
28	101	11	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
29	101	11	25400	1835	0	0	0	0	0	172.7	0	23390	8117	15280	0	0
30	101	11	0	188.8	4058	0	56300	0	0	-188.8	0	0	-4058	-56300	0	0
1	201	12	25400	5371	0	0	65070	0	0	-50.29	0	20080	20080	-65070	0	0
2	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
3	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
4	201	12	12000	3856	0	0	0	0	0	-2768	0	10910	10910	0	0	0
5	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
6	201	12	22000	4716	0	0	0	0	0	-44.17	0	17330	11180	6144	0	0
7	201	12	9000	3049	0	0	0	0	0	-2421	0	8371	0	8371	0	0
8	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
9	201	12	10000	3255	0	0	0	0	0	-2423	0	9168	4058	5109	0	0
10	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
11	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
12	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
13	201	12	1000	0	4058	0	0	0	0	0	0	1000	-3058	0	0	0
14	201	12	20000	4033	0	0	0	0	0	-276.3	0	16240	15230	1010	0	0
15	201	12	3000	0	4058	0	0	0	0	0	0	3000	-1058	0	0	0
16	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
17	201	12	15000	2777	0	0	0	0	0	-375.9	0	12600	5117	7482	0	0
18	201	12	0	0	4058	0	55830	0	0	0	0	0	-4058	-55830	0	0
19	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0

Table A-1. Continued

DAY	Node	Process	Precipi- tation	Runoff	Evapo- transpir- ation	Seep- age	BaseF (TF)	Base- flow	Ground- water Recharge	Flows in excess of 24 hours	De- tention	Infil- tration	Root zone change in soil moisture	Subsurface change in soil moisture	Total error	Percent Error
20	201	12	15000	2699	0	0	0	0	0	-375.2	0	12680	8117	4560	0	0
21	201	12	7000	0	0	0	0	0	0	0	0	7000	0	7000	0	0
22	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
23	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
24	201	12	4000	0	0	0	0	0	0	0	0	4000	4000	0	0	0
25	201	12	13000	2183	0	0	0	0	0	-370.2	0	11190	4117	7070	0	0
26	201	12	15000	2912	0	0	0	0	0	-377.2	0	12470	0	12470	-1.72	-0.01
27	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
28	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0
29	201	12	25400	6191	0	0	0	0	0	-57.83	0	19270	8117	11150	0	0
30	201	12	0	0	4058	0	0	0	0	0	0	0	-4058	0	0	0

APPENDIX B
STREAMGAGE SITE DETAILS

Figure B-1. Timau – CA01 streamgage from downstream looking upstream. This is after a car accident occurred so gauge housing is destroyed at the top. The flow at this section of the stream is typically low and heavily altered due to the culvert and bridge infrastructure present, and the riverbed is primarily bedrock in this section. Due to this, the streamgauge was only able to monitor flows effectively during large rainfall pulses. Photo courtesy of author.

Figure B-2. Timau – CA02 streamgauge from downstream right bank looking upstream. Similar to CA01, this gauge location is only responsive to high rainfall and installation was limited by bedrock. Photo courtesy of author.

Figure B-3. Storms Bridge – CA03 streamgage from downstream looking upstream. This gauge is subject to very ‘flashy’ streamflows and is shown here during low-flow conditions. Photo courtesy of author.

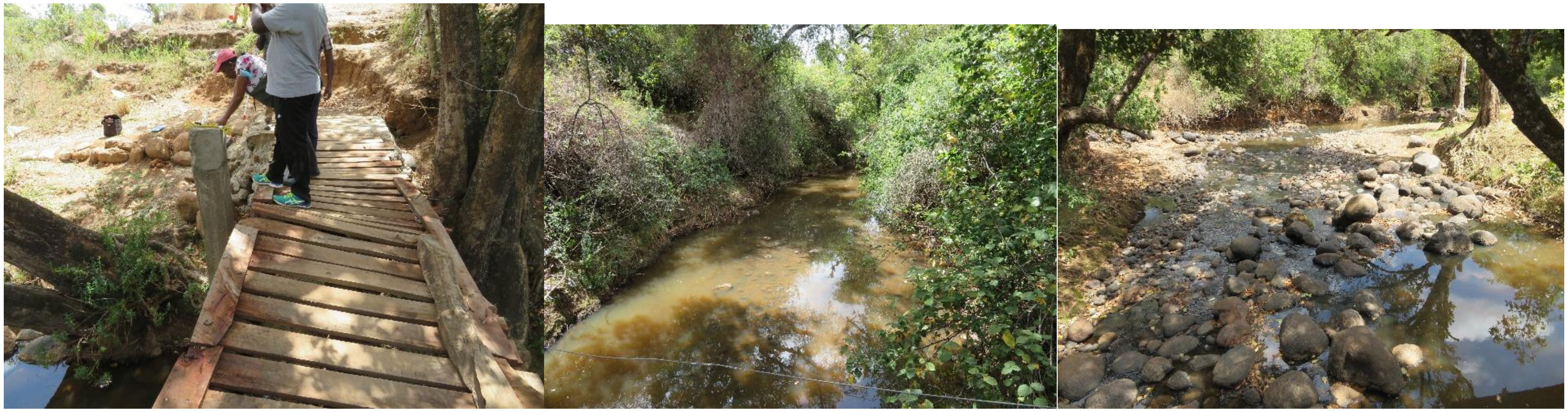

Figure B-4. Likii – CA04 streamgage (left) with upstream (center) and downstream (right) views. This streamgage was destroyed during a high flow event and large debris that was carried through the channel during the event. During low flows, there is typically still a deep pool at the gauge location. Photos courtesy of author.

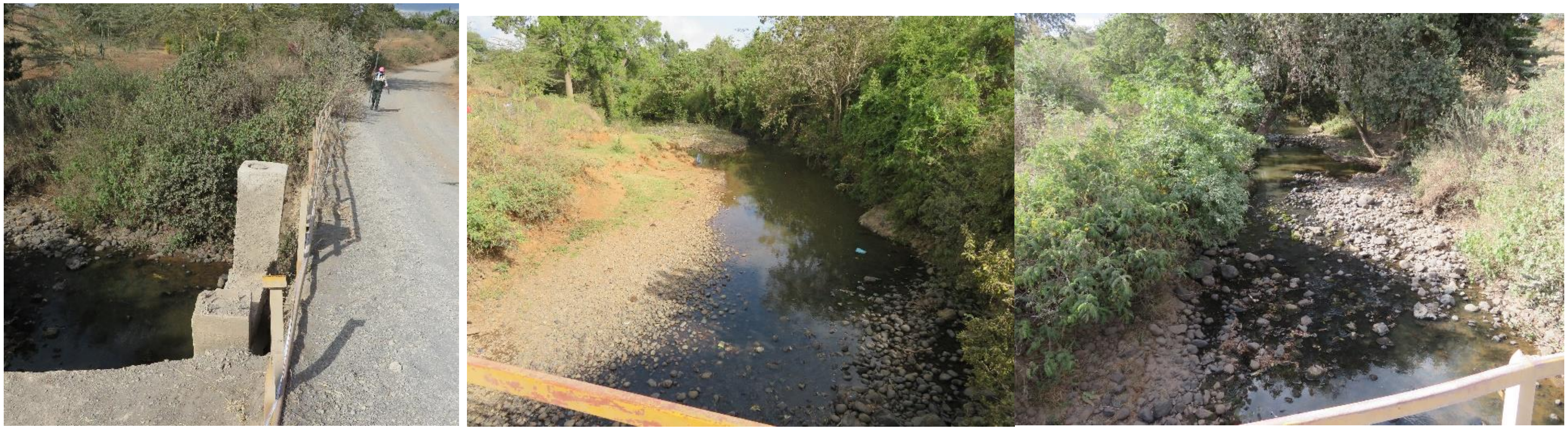

Figure B-5. Mukima – CA05 streamgauge (left) with upstream (center) and downstream (right) views. This gauge also sits at a somewhat deep pool due to the bridge and road infrastructure it is co-located with. Photos courtesy of author.

Figure B-6. Juakali – CA06 streamgage during low/no flow conditions. This gauge is located at a popular stream access point for the community, and therefore the banks and geometry at this location are heavily influenced by both the road infrastructure and the widened and less steep banks that have been carved out through access pathways. Once again, a pool is present under no-flow conditions. Photo courtesy of author.

Figure B-7. Doldol – CA07 streamgage during low/no flow conditions. The stream channel is partially bedrock constricted and the gauge (located to the far right) is located in a pool even when there is no flow. Photo courtesy of author.

Figure B-8. Ol Jogi – CA08 streamgage during low/no-flow (left), looking upstream (center) and downstream (right). Looking closely the bottom of the gauge housing, the lateral pipe is visible above the water. Once again, the bedrock present at the site made installation difficult to capture very low flows, but at current levels the river barely has any flow. Photos courtesy of author.

APPENDIX C
RATING CURVE DETAILS

Table C-1. Rating curves developed for each site, including low and high flow equations as needed.

Site ID	Name	Primary Equation	Range of depths (m)	R2	Estimate Flow =0 depth	Low Flow Equation	Range of Depths	R2	High Flow equation	Range of depths	R2
CA01	Timau 0125	NA									
CA02	Timau 08	$22.019x - 0.12$	>0.005	0.46	0.005						
CA03	Storms Bridge	$4.1355x-1.4055$	.357-0.65	0.895	0.2	$y = 0.5342x - 0.1162$	.20-0.357	0.78	$y = 14.558x - 8.1858$	>0.65	0.98
CA04	Likii	$6.8822x-1.8669$	0.293 - 0.671	0.9153	0	$y = 0.5505x$	<0.293				
CA05	Mukima	$8.8824x-2.7064$	0.359-0.922	0.8703	0.216	$y = 16.031x^2 - 5.8471x + 0.5156$	.216-.359	1.00			
CA06	Juakali	$9.1536x-3.5509$	0.388-1.00	0.8225	0.388	$y=0$	<0.388				
CA07	Doldol	$10.822x - 5.159$	>0.5	0.7336	0	$y = 7.1545x^{4.8412}$	0-0.5	0.95			
CA08	Ol Jogi	$4.2189x - 1.3163$	>0.48	0.6263		$y = 0.1784x + 0.6531$	$y = 1.4636x + 0.0072$	0.05			

APPENDIX D
LAIKIPIA SUBWATERSHED PHYSICAL PROPERTIES

Table D-1. Laikipa, Kenya subwatershed physical characteristics.

Node	CN	Area,ha	longest flow path, m	Watershed Slope, m/m	Soil Texture (#)	om	uFC	uWP	Soil porosity	VsatK	Sav	WCSat
101	70.3	845	11747	0.07	5	3.8	0.35	0.22	0.626	0.18	48.06	0.63
103	84.4	27	579	0.04	5	2.4	0.38	0.25	0.592	0.11	53.75	0.59
104	81	2069	12984	0.05	5	2.5	0.37	0.24	0.586	0.13	50.98	0.59
105	77.6	742	7231	0.03	5	1.9	0.36	0.23	0.578	0.14	47.00	0.58
107	83	7013	26595	0.04	5	2.3	0.37	0.24	0.587	0.13	49.09	0.59
108	80.3	9419	19332	0.06	5	2.2	0.32	0.20	0.563	0.26	40.34	0.56
110	85.1	4404	18751	0.03	5	2.1	0.36	0.23	0.579	0.15	43.83	0.58
112	85.6	88	1371	0.02	5	1.8	0.35	0.23	0.554	0.13	41.02	0.55
113	88.2	80	1476	0.02	1	1.8	0.37	0.25	0.544	0.09	43.06	0.54
115	79.7	5461	15849	0.04	6	1.9	0.30	0.19	0.533	0.34	34.03	0.53
117	86.3	105	1766	0.03	5	1.8	0.36	0.24	0.522	0.12	42.97	0.52
118	83.7	12362	24336	0.04	6	1.9	0.29	0.18	0.526	0.38	34.28	0.53
119	85.9	4796	12475	0.02	5	1.8	0.33	0.21	0.520	0.23	39.62	0.52
201	58.7	2877	12414	0.08	5	4	0.34	0.21	0.627	0.19	45.41	0.63
202	85.3	101	1496	0.04	5	2.6	0.36	0.23	0.599	0.14	51.90	0.60
301	65.2	2899	11470	0.08	5	3.7	0.34	0.22	0.584	0.19	44.97	0.58
302	85.5	852	7681	0.05	1	2.6	0.38	0.25	0.591	0.11	53.37	0.59
303	85.1	154	3241	0.03	1	2.6	0.40	0.27	0.592	0.08	52.11	0.59
401	51.9	5641	24238	0.11	5	6	0.34	0.22	0.632	0.18	43.34	0.63
402	85.6	459	5093	0.06	5	3.1	0.38	0.24	0.607	0.12	52.51	0.61
403	87.6	1226	10998	0.04	1	2.2	0.39	0.26	0.588	0.10	51.06	0.59
405	89.1	735	7827	0.02	5	2.3	0.37	0.24	0.595	0.11	45.26	0.59

Table D-1. Continued

Node	CN	Area,ha	longest flow path, m	Watershed Slope, m/m	Soil Texture (#)	om	uFC	uWP	Soil porosity	VsatK	Sav	WCSat
501	74.6	1014	6827	0.07	5	3.3	0.35	0.23	0.563	0.14	44.66	0.56
502	86.4	2301	15705	0.03	1	2.3	0.38	0.25	0.591	0.10	50.93	0.59
601	71.2	2650	14016	0.08	5	4.5	0.35	0.22	0.577	0.16	42.68	0.58
602	71.1	3588	26891	0.09	5	4	0.36	0.23	0.607	0.14	46.53	0.61
603	84.9	4364	30471	0.02	5	2.1	0.37	0.24	0.580	0.11	48.05	0.58
701	67.3	6109	29224	0.11	5	4.5	0.35	0.22	0.595	0.16	42.00	0.60
703	85.7	3025	11483	0.03	5	2.1	0.37	0.24	0.583	0.12	49.04	0.58
704	90.8	809	8454	0.02	5	1.7	0.37	0.24	0.563	0.10	45.03	0.56
706	88.6	186	2580	0.02	5	1.7	0.36	0.24	0.586	0.11	43.06	0.59
707	85.6	962	8626	0.01	5	1.8	0.37	0.24	0.559	0.10	42.22	0.56
709	88.4	69	1404	0.02	5	1.7	0.36	0.24	0.540	0.11	43.23	0.54
710	88.5	215	3345	0.02	5	1.7	0.37	0.24	0.538	0.10	43.25	0.54
801	68.8	8417	31710	0.12	5	4.7	0.34	0.22	0.608	0.18	41.99	0.61
901	64.9	5215	26368	0.14	5	6	0.33	0.21	0.627	0.21	40.45	0.63
902	79.4	1731	11744	0.02	5	2	0.37	0.24	0.569	0.12	48.63	0.57
903	88.7	405	4556	0.01	5	1.9	0.36	0.24	0.556	0.11	43.27	0.56
1000	86.3	5052	27962	0.03	5	2.1	0.37	0.24	0.583	0.11	46.82	0.58
1100	83.1	4381	21089	0.01	5	1.8	0.37	0.24	0.538	0.11	42.59	0.54

APPENDIX E
SAI SCENARIO INPUTS

Table E-1. Irrigation fractions depending on scenario (H= high rainfall area, M = high rainfall area, and H&M = entire watershed at 50%).

Node	Percent Ag	Rainfall Zone Classification	H Scenarios Irrigation Fraction	M Scenarios Irrigation Fraction	H&M Scenarios Irrigation Fraction
101	61	High	0.12	0	0.06
103	92	High	0.18	0	0.09
104	70	High	0.14	0	0.07
105	55	High	0.11	0	0.055
107	85	High	0.17	0	0.085
108	13	Medium	0	0.03	0.015
110	58	Medium	0	0.12	0.06
112	57	Medium	0	0.11	0.055
113	82	Medium	0	0.16	0.08
201	32	High	0.06	0	0.03
202	91	High	0.18	0	0.09
301	13	High	0.03	0	0.015
302	89	High	0.18	0	0.09
303	83	High	0.17	0	0.085
402	97	High	0.19	0	0.095
403	96	High/medium	0	0	0.095
405	94	Medium	0	0.19	0.095
501	10	High	0.02	0	0.01
502	96	High/medium	0	0	0.095
601	6	High	0.01	0	0.005
602	49	High	0.1	0	0.05
603	71	High/medium	0	0	0.07
703	89	High	0.18	0	0.09
704	65	Medium	0	0.13	0.065
706	76	Medium	0	0.15	0.075
707	68	Medium	0	0.14	0.07
709	84	Medium	0	0.17	0.085
710	77	Medium	0	0.15	0.075
801	8	High	0.02	0	0.01
901	8	High	0.02	0	0.01
902	58	High/medium	0	0	0.06
903	49	Medium	0	0.1	0.05
1000	77	High/medium	0	0	0.075
1100	20	Medium	0	0.04	0.02

Table E-2. Curve number values for different tillage scenarios.

Node	NT H 100	NT M100	RT H 100	RT M 100	NT H&M 100	RT H&M 100	NT H&M 50	RT H&M 50
101	66.4	70.3	67.3	70.3	66.4	67.3	68.4	68.8
103	77.4	84.4	79.0	84.4	77.4	79.0	80.9	81.7
104	75.9	81.0	77.0	81.0	75.9	77.0	78.4	79.0
105	73.7	77.6	74.6	77.6	73.7	74.6	75.7	76.1
107	76.6	83.0	78.0	83.0	76.6	78.0	79.8	80.5
108	80.3	80.3	80.3	79.6	79.4	79.6	79.8	79.9
110	85.1	85.1	85.1	81.7	80.7	81.7	82.9	83.4
112	85.6	85.6	85.6	82.2	81.2	82.2	83.4	83.9
113	88.2	88.2	88.2	83.2	81.7	83.2	85.0	85.7
201	57.0	58.7	57.4	58.7	57.0	57.4	57.9	58.0
202	78.3	85.3	79.9	85.3	78.3	79.9	81.8	82.6
301	64.5	65.2	64.6	65.2	64.5	64.6	64.8	64.9
302	78.6	85.5	80.2	85.5	78.6	80.2	82.1	82.8
303	78.7	85.1	80.1	85.1	78.7	80.1	81.9	82.6
402	78.1	85.6	79.8	85.6	78.1	79.8	81.9	82.7
403	83.8	83.8	84.7	84.7	80.0	81.7	83.8	84.7
405	89.1	89.1	89.1	83.2	81.5	83.2	85.3	86.2
502	81.0	84.3	82.2	84.8	78.9	80.6	82.7	83.5
602	68.0	71.1	68.7	71.1	68.0	68.7	69.5	69.9
603	82.0	82.3	82.7	82.9	79.4	80.7	82.2	82.8
703	78.9	85.7	80.4	85.7	78.9	80.4	82.3	83.0
704	90.8	90.8	90.8	86.7	85.5	86.7	88.2	88.7
706	88.6	88.6	88.6	83.9	82.6	83.9	85.6	86.3
707	85.6	85.6	85.6	81.5	80.4	81.5	83.0	83.6
709	88.4	88.4	88.4	83.2	81.7	83.2	85.1	85.8
710	88.5	88.5	88.5	83.7	82.3	83.7	85.4	86.1
902	76.4	78.3	77.0	78.5	75.2	76.2	77.3	77.8
903	88.7	88.7	88.7	85.6	84.8	85.6	86.7	87.2
1000	83.6	83.1	84.2	83.8	80.3	81.7	83.3	84.0
1100	83.1	83.1	83.1	81.9	81.6	81.9	82.3	82.5

APPENDIX F
CUENCA INPUT FILES

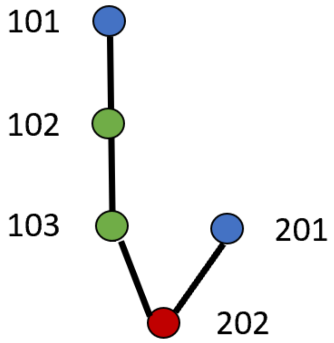

Figure F-1. Link and node configuration of example file. Blue circles correspond to nodes that only have rainfall-runoff calculations, green nodes correspond to nodes that have both rainfall runoff and convex channel routing, and red nodes correspond to "add" processes, where the hydrograph from node 201 is added to node 103 and is retained in Stream 2 through node 202.

The following is an example of input configuration file (.idat). The first three values for each process correspond to Node 1 (NZ1), Node 2 (NZ2), and the process number (Kode) in the CUENCA program. The remaining inputs can be cross referenced in the Code in Appendix G in files UHCN.f for rainfall-runoff, convex.f, and add.f.

```
101,101,11     ← Upstream node, downstream node, and process code (UHCN.f)
1,67.1,100,4,24,2000,0.03,1,-1,1,1,-1,1,2,0.42,0.42,.3,0.5,0.5,0.5,0.482,2,0,0,0 ←
                                                         process arguments
101,102,5 ← convex process (convex.f)
1,0.3,1,1,24,10,2,1000,998,100,0.05,100,0,0,0,0.001,1.52
102,103,11 ← Curve number process (UHCN.f)
```

```
1,67.1,100,4,24,2000,0.03,1,-1,1,1,-1,1,2,0.42,0.42,.3,0.5,0.5,0.5,0.482,2,0,0,0
102,103,5  ← convex process (convex.f)
1,0.3,1,1,24,10,2,1000,998,100,0.05,100,0,0,0,0.001,1.52
201,201,11← Curve number process (UHCN.f)
1,67.1,100,4,24,2000,0.03,1,-1,1,1,-1,1,2,0.42,0.42,.3,0.5,0.5,0.5,0.482,2,0,0,0
103,202,7 ← add process (add.f)
1,2,201,1
999,999,999   ← signifies end of process inputs
```

The following is an example of a precipitation input file.

```
9    6    1    'NRDAYS    NRNODES PRINTSELECTION'
     'Precip mm
0    101  102  103  201
152  25.4 25.4 25.4 25.4
153  0    0    0    0
154  0    0    0    0
155  12   12   12   12
156  0    0    0    0
157  22   22   22   22
158  9    9    9    9
159  0    0    0    0
160  10   10   10   10
```

The following is an example of an output file for daily flow.

```
File: output/uhcngaT64.odss                 CUENCA v0.2, 10/2022
          >>>>> DAILY FLOW VOLUME (M^3) <<<<<

   DAY         101         102         103         201         202
    1    0.50057E+03 0.76685E+03 0.28491E+03 0.50208E+03 0.23781E+03
    2    0.39111E+02 0.76685E+03 0.17724E+02 0.50208E+03 0.16417E+02
    3    0.23263E+02 0.76685E+03 0.00000E+00 0.50208E+03 0.00000E+00
    4    0.23263E+02 0.76685E+03 0.00000E+00 0.50208E+03 0.00000E+00
    5    0.23263E+02 0.76685E+03 0.00000E+00 0.50208E+03 0.00000E+00
    6    0.23263E+02 0.76685E+03 0.00000E+00 0.50208E+03 0.00000E+00
    7    0.23263E+02 0.76685E+03 0.00000E+00 0.50208E+03 0.00000E+00
    8    0.23263E+02 0.76685E+03 0.00000E+00 0.50208E+03 0.00000E+00
    9    0.23263E+02 0.76685E+03 0.00000E+00 0.50208E+03 0.00000E+00
```

APPENDIX G
CUENCA MODEL CODE

```
C      PROGRAM 18  ! Based on Hromadka book pag 222
C --------------------------------------------------------------
-----------
      SUBROUTINE ADD(DAYQI,DAYQO)
C --------------------------------------------------------------
-----------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCC
C THIS SUBROUTINE ADDS A STREAM DATA BANK TO ANOTHER STREAM DATA
BANK     C
C VARIABLES:
C
C NUMA:  Streams to be added
C
C NUMS:  Stream to receive flow from another stream
C
C NODA:  Node to be added (i.e. corresponding to stream A)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCC
C --------------------------------------------------------------
-----------
C   DECLARE VARIABLES
C --------------------------------------------------------------
------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C       PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/INPUTS/DATAINP(100,100)
      DIMENSION A(5555),B(5555),DAYQO(5555,100),DAYQI(5555,100)
      DIMENSION
SUMQO24(5555),SUMA24(5555),SUMB24(5555),SUMQI24(5555)
C --------------------------------------------------------------
--------------
C READ INPUTS
C---------------------------------------------------------------
--------------
      NUMA=INT(DATAINP(JCOUNT,4)) !stream 2
      NUMS=INT(DATAINP(JCOUNT,5)) !stream 1
      NODA=INT(DATAINP(JCOUNT,6)) !
      AREA=DATAINP(JCOUNT,7)
```

```
      SUMA24=0.D0
      SUMB24=0.D0
      SUMQI24=0.D0
      SUMQO24=0.D0
C -----------------------------------------------------------------
--------------
C READ FLOW VALUES FROM STREAMS
C-----------------------------------------------------------------
--------------
      CALL MREAD(NUMS,A) !stream 1
      CALL MREAD(NUMA,B) !stream 2
      NUMBS=INT(A(5555))
      NUMBA=INT(B(5555))
      NUMBER=NUMBS
C --- Calculate and write inflow volume in m3 to storage matrix
      SUMA24(1)=0.5d0*A(1)*5.d0*60.d0*0.0283168d0
      SUMB24(1)=0.5d0*B(1)*5.d0*60.d0*0.0283168d0
      SUMQI24(1)=SUMA24(1)+SUMB24(1)
      DO 10 I=2,288
          SUMA24(I)=SUMA24(I-1)+0.5d0*(A(I-1)+
     &                A(I))*5.d0*60.d0*0.0283168d0 !Calculating
the sum by taking average of timesteps, multiplying over time
step to get volume, and converting to cubic meters
          SUMB24(I)=SUMB24(I-1)+0.5d0*(B(I-1)+
     &                B(I))*5.d0*60.d0*0.0283168d0
          SUMQI24(I)=SUMA24(I)+SUMB24(I)
10    CONTINUE
      DO 20 J=1,100
            IF (INT(DPRECIP(1,J)).EQ.NZ2) THEN
               INODE=J
            END IF
20    CONTINUE
      DAYQI(JDAY,INODE)=SUMQI24(288) ! Daily total inflow volume
is equal to sum at timestep 288

      IF(NUMBA.GT.NUMBS)NUMBER=NUMBA
      NUMBER = 5555
      !IF (NUMBER.GT.0.D0) THEN
       DO 100 I=1,NUMBER-1
           A(I)=A(I)+B(I)
100     CONTINUE
        SUMQO24(1)=0.5d0*A(1)*5.d0*60.d0*0.0283168d0
        DO 120 I=2,288
          SUMQO24(I)=SUMQO24(I-1)+0.5d0*(A(I-1)+
     &        A(I))*5.d0*60.d0*0.0283168d0
120     CONTINUE
          DAYQO(JDAY,INODE)=SUMQO24(288)
```

```
        !DAYQO(JDAY,INODE) = SUMQI24(288)
       !A(5555)=NUMBER
       CALL MWRITE(NUMS,A)
      !ELSE
      !    WRITE(nut,999)
      !END IF
C --------------------------------------------------------------
-----------------
C OUTPUT - FORMAT
C --------------------------------------------------------------
-----------------
      WRITE(NUT,101)NUMA,NUMS
101   FORMAT(10X,'STREAM NUMBER',I2,' ADDED TO STREAM
NUMBER',I2)
999   FORMAT(10X,'NO FLOW OCCURRED ON THIS DAY')
C --------------------------------------------------------------
-----------------
C END PROCESS
C --------------------------------------------------------------
-----------------
      RETURN
      END SUBROUTINE ADD
```

```
!  Part of FLOOD Program 15 (old main from the book) - Based on
Hromadka book pag 197
C  -------------------------------------------------------------
---------
      SUBROUTINE ADDHY(UNIT,INTERV,NA,H)
C  -------------------------------------------------------------
---------
C     *USED FOR CUENCA ROUTING SYSTEM ONLY*
C     TRANSFORMS SUBROUTINE UNITH FOR USE WITH THE FLOOD SYSTEM
C     JUST INSERT CALL TO ADDHY AT END OF SUBROUTINE UNITH
C     IMMEDIATELY PRECEEDING CALL TO OABS
C     ADD RUNOFF HYDROGRAPH TO A STREAM
C --------------------------------------------------------------
---------
C   DECLARE VARIABLES
C --------------------------------------------------------------
---------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      DIMENSION H(5555)
      DIMENSION AA(5555)
C --------------------------------------------------------------
---------
      CALL MREAD(NA,AA)
      NUMX=INT(UNIT/5.d0+.01d0)
C --------------------------------------------------------------
---------
      IF(NUMX-2)751,752,753
751   DO 750 I=1,INTERV
         AA(I)=AA(I)+H(I)
750   CONTINUE
      GO TO 760
752   DO 755 I=1,INTERV
          J=2*I
          K=J-1
          AA(K)=AA(K)+H(I)
          AA(J)=AA(J)+H(I)
755   CONTINUE
      GO TO 760
753   DO 756 I=1,INTERV
         L=3*I
         K=L-1
         J=L-2
         AA(L)=AA(L)+H(I)
         AA(K)=AA(K)+H(I)
         AA(J)=AA(J)+H(I)
756   CONTINUE
```

```
760   AA(5555)=INTERV*NUMX
      CALL MWRITE(NA,AA)
C --------------------------------------------------------------
---------
      RETURN
      END SUBROUTINE ADDHY
```

```
C ----------------------------------------------------------------
--------
C  PROGRAM:  SCS concentration time calculation
C ----------------------------------------------------------------
--------------------
       SUBROUTINE calctc(pL,CN,Y,tc)
C ----------------------------------------------------------------
--------------------
C      version 3.0.1, Last ModIFied: See ModIFications below
C      WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C      Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons,
BAE (jep)
C                  University of Florida        BAE, NC State
University
C                  Gainesville, FL 32611        Raleigh, NC
27695-7625(USA)
C                  e-mail: carpena@ufl.edu
C ----------------------------------------------------------------
-------------------
C   DEFINE VARIABLES
C     PL: Longest flow path, m
C     CN: Curve number, dimensionless
C     Y: Watershed slope, m/m
C     tc: time of concentration, hours
C ----------------------------------------------------------------
-------------------
C   DECLARE VARIABLES
C ----------------------------------------------------------------
-------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
C ----------------------------------------------------------------
-------------------
      tc=pL**0.8d0*(1000.d0/CN-9.d0)**0.7d0/(4407.d0*dsqrt(Y))

      RETURN
      END SUBROUTINE calctc
```

```
CC PROGRAM 18  -  Based on Hromadka book pag 222
C --------------------------------------------------------------
---------
      SUBROUTINE CLEAR(DAYQI,DAYQO,DAYMO) ! ARGU = nut
C --------------------------------------------------------------
---------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCC
C THIS SUBROUTINE CLEARS A SPECIFIED STREAM DATA BANK
C
C Variables:
C
C
C
C K: Stream number to be set to 0
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCC
C --------------------------------------------------------------
----------
C   DECLARE VARIABLES
C --------------------------------------------------------------
----------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/INPUTS/DATAINP(100,100)
      DIMENSION A(5555),DAYQI(5555,100),DAYQO(5555,100),
     & DAYMO(5555,100),SUMQI24(5555)
!     EXPORT Hydrograph, Date (hours) StreamA(CFS)
C --------------------------------------------------------------
----------
!      SS=SS1
!    CLEAR THE K STREAM IN STREAM MATRIX SS
      !READ(nut,*)K
      K=INT(DATAINP(JCOUNT,4))
      AREA=DATAINP(JCOUNT,5)
      SUMQI24=0.D0
      CALL MREAD(K,A)
C --- Calculate and write inflow volume (in mm) to storage
matrix
      SUMQI24(1)=0.5D0*A(1)*5.D0*60.D0*0.0283168D0
      DO 10 I=2,288
          SUMQI24(I)=SUMQI24(I-1)+0.5d0*(A(I-1)+
```

```
     &                A(I))*5.d0*60.d0*0.0283168d0 !Calculating
the sum by taking average of timesteps, multiplying over time
step to get volume, and converting to cubic meters
10    CONTINUE
      DO 20 J=1,100
            IF (INT(DPRECIP(1,J)).EQ.NZ2) THEN
               INODE=J
            END IF
20    CONTINUE
      DAYQI(JDAY,INODE)=SUMQI24(288) ! Daily inflow volume is
equal to the sum at timestep 288
      DO 30 I=1,5555
         A(I)=0.D0
30    CONTINUE
      DAYMO(JDAY,INODE)=DAYQI(JDAY,INODE)
      DAYQO(JDAY,INODE)=0.D0
      CALL MWRITE(K,A)
C ----------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C ----------------------------------------------------------------
----------
      WRITE(NUT,101)K
101   FORMAT(10X,'STREAM NUMBER',I2,' IS SET TO ZERO.')

      RETURN
      END SUBROUTINE clear
```

```
C      PROGRAM 19 - Based on Hromadka book 231 pag
C  ----------------------------------------------------------------
-----------------
      SUBROUTINE
convex(DAYQI,DAYQO,DAYDN,DAYDS,DSEEP,DSPRING,DSNOW, !ARGU = nut
     &                DBASEF,DRECH,DSM2,SISTORE,DTHETA2,
     &
dstorvol,dmaxstor,drloss,dSIWater1,dminstor,
     &                dirreff)
C  ----------------------------------------------------------------
-----------------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCC
C THIS SUBROUTINE MODELS CHANNEL ROUTING BY THE SIMPLER CONVEX
METHOD          C
C WHERE A CSTAR VALUE AS ESTIMATED DUE TO IRREGULAR DELT VALUES
C
C Variables:
C
C ***** Line 1: 'NA,C,V0,TIME1,TIME2' *****
C
C NA:     Stream 'A" number. This stream is the one to be
modeled                  C
C C:      Channel routing coefficient [0.01 – 1.0].
C
C V0:     Channel average flow velocity (m/s) [0.003-30]
C
C TIME1:  Time for Beginning of results (hrs)
                     C
C TIME2:  Time for End of results (hrs)
                    C
C ***** Line 2: 'BB,Z,E1,E2,XL,XN' *****
                         C
C BB:     Base width (m). Allowable values [0.003-300]
                    C
C Z:      Channel “Z” factor – Ratio of Horizontal/vertical. [0
- 100]              C
C E1:     Upstream elevation (m) [-3  to 3000]
                         C
C E2:     Downstream elevation (m) [-60 to 3000]
                         C
C XL:     Channel length  - the length of the  longest
watercourse (m)              C
C XN:     Basin Factor (Manning's Friction Factor) [0.008 -
0.999]                   C
C Satk: Hydraulic conductivity of streambed (cm/hr)
                    C
```

```
C hc: Thickness of clogging layer at bottom of stream (m)
               C
C hg: Groundwater head (m)
               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCC
C     INTERNAL CALCULATION VARIABLES
C  Hriv: Head of river (ft)
C ----------------------------------------------------------------
-------------------
C   DECLARE VARIABLES
C ----------------------------------------------------------------
-------------------
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK10/B(5555)
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      DIMENSION
dstorvol(100),dmaxstor(100),dminstor(100),drloss(100),
     & dSIWater1(100),DIRREFF(5555,100)
      COMMON/INPUTS/DATAINP(100,100)
      DIMENSION DAYQI(5555,100),DAYQO(5555,100),DSEEP(5555,100),
     &
DAYDN(5555,100),DAYDS(5555,100),SUMQI24(5555),SUMA24(5555),
     &
SEEP24(5555),DSPRING(5555,100),DSNOW(5555,100),DBASEF(5555,100),
     &
DRECH(5555,100),DSM2(5555,100),SISTORE(100),DTHETA2(5555,100)
      DIMENSION A(5555),AA(5555),CC(5555),DD(5555),S(5555)

common/CINPUT/DETO(5555,100),DBF(5555,100),DTAVG(5555,100),
     &
DTMAX(5555,100),DTMIN(5555,100),DWS2(5555,100),DSORAD(5555,100),
     &
DCKM(5555,100),DAB(5555,100),DIRR(5555,100),DSNO(5555,100)
!     EXPORT Hydrograph, nute (hours) StreamA(CFS)
C ----------------------------------------------------------------
---------------------
C INITIALIZE VARIABLES
C ----------------------------------------------------------------
--------------------
      TIME=0.d0
!      SS=SS1
```

```
C -----------------------------------------------------------------
--------------------
C INPUT DATA
C -----------------------------------------------------------------
--------------------
      NA=INT(DATAINP(JCOUNT,4))
      C=DATAINP(JCOUNT,5)
      V0=DATAINP(JCOUNT,6)
      TIME1=DATAINP(JCOUNT,7)
      TIME2=DATAINP(JCOUNT,8)
      BB=DATAINP(JCOUNT,9)
      Z=DATAINP(JCOUNT,10)
      E1=DATAINP(JCOUNT,11)
      E2=DATAINP(JCOUNT,12)
      XL=DATAINP(JCOUNT,13)
      XN=DATAINP(JCOUNT,14)
      AREA=DATAINP(JCOUNT,15)
      satk=DATAINP(JCOUNT,16)
      hc=DATAINP(JCOUNT,17)
      hg=DATAINP(JCOUNT,18)
      arec=DATAINP(JCOUNT,19)
      brec=DATAINP(JCOUNT,20)
      SUMA24=0.D0
C      SUMB24=0.D0
      SUMQI24=0.D0
C      SUMQO24=0.D0
      SEEP24=0.D0
      SPRING24=0.D0
      dirrcfs=0.d0
C -----------------------------------------------------------------
--------------------
C CONVERSION
C -----------------------------------------------------------------
--------------------
      v0=v0/0.3048d0 !to convert m to feet
      bb=bb/0.3048d0 !to convert m to feet
      e1=e1/0.3048d0 !to convert m to feet
      e2=e2/0.3048d0 !to convert m to feet
      xl=xl/0.3048d0 !to convert m to feet
      hc=hc/0.3048d0 !to convert m to feet
      hg=hg/0.3048d0 !to convert m to feet
      satk=satk/2.54d0/3600.d0/12.d0 ! convert cm/hr to ft/s
C -----------------------------------------------------------------
--------------------
      WRITE(NUT,901)NA
      IF(C.GT.0.d0)WRITE(NUT,903)C
      IF(V0.GT.0.d0)WRITE(NUT,904)V0
```

```
      WRITE(NUT,905)BB,Z,E1,E2,XL,XN,satk,hc,hg
C --------------------------------------------------------------
--------------------
      DO 5 I=1,5555
         B(I)=0.d0
         S(I)=0.d0
5     CONTINUE
      DO 8 J=1,100
            IF (INT(DPRECIP(1,J)).EQ.NZ2) THEN
               INODE=J
            END IF
            IF (INT(DPRECIP(1,J)).EQ.NZ1) THEN
               JNODE=J
            END IF
8     CONTINUE
      !CALL MREAD(NA,A)
      CALL NREAD(NZ1,AA) !read values from upstream node to be
routed through stream (cfs)
      CALL NREAD(NZ2,CC) !read values from current node (i.e.
any rainfall-runoff process that has occurred) (CFS)
      DO 9 J=1,5555      !values from upstream node are written
to A() matrix
          A(J)=AA(J)
9     CONTINUE
      ISUM=0.d0
      QSUM=0.d0
      NUMBER=INT(A(5555))
      NUMBER = 0.D0 !!! lw TESTING 5.1.2023
      SNO=DSNO(JDAY+1,iNODE)*35.317d0 !add snowmelt/baseflow
(convert from m3/s to cfs)
      !print*,'sno',sno

      BF=((DBF(JDAY,JNODE)+DBASEF(JDAY,JNODE))/(3600.d0*24.d0))!
convert baseflow contribution from Rainfall-RO (m3/day) to ft3/s
      DIRRCFS=DIRREFF(JDAY,INODE)*35.3147D0/(3600.d0*24.d0)
!convert daily m3 irrigation withdrawal to cfs
C --- ADD baseflow from node just upstream of convex section,
then subtract irrigation removals, recalculate "number"
      DO 10 I=1,5555-1
          IF (I.le.288) then
               A(I) = A(I) + BF ! Add daily baseflow
contribution and
               !PRINT*,'a(I) AFTER BASEFLOW',a(I)
          end if
10    CONTINUE
      IF (DIRRCFS.GT.0.D0) THEN
          DO 40 J=1,5555
```

```
          AA(J)=A(J)
          !PRINT*,'AA,A',AA(J),A(J)
40     CONTINUE
          EXCESS=0.D0
          DO 50 K=1,288
              DIRRCFS=EXCESS+DIRRCFS
              !print*,'excess',excess
               A(K) = AA(K) - DIRRCFS
               !print*,'aa,dirrcfs,a',AA(K),diRRcfs,A(K)
               IF (A(K).LT.0.D0) THEN
                   EXCESS=DIRRCFS-AA(K)
                   A(K)=0.D0
               END IF
50        CONTINUE
      END IF
          DO 60 J=1,5555-1
          IF (A(J).GT.0.D0) THEN
             NUMBER = NUMBER + 1
         ELSE
             NUMBER = NUMBER
         END IF
60    CONTINUE
      A(5555)=NUMBER
C --- Calculate and write inflow volume (in m3) to storage
matrix
      SUMQI24(1)=0.5D0*A(1)*5.D0*60.D0*0.0283168D0
      DO 70 I=2,288
          SUMQI24(I)=SUMQI24(I-1)+0.5d0*(A(I-1)+
     &        A(I))*5.d0*60.d0*0.0283168d0 !Calculating the sum
by taking average of timesteps, multiplying over time step to
get volume, and converting to cubic meters
70    CONTINUE

      DAYQI(JDAY,INODE)=SUMQI24(288)  ! Daily inflow volume is
equal to the sum at timestep 288
C --- Determine if any flow is in hydrograph ---
      !IF (NUMBER.GT.0.D0) THEN
C ----------------------------------------------------------------
--------------------
C FIND QMAX OF STREAM
C ----------------------------------------------------------------
--------------------
       QMAX=0.d0
       !DO 13 I=1,NUMBER
       DO 80 I=1,5555-1
           IF(A(I).GT.QMAX)QMAX=A(I)
80    CONTINUE
```

```
       F=QMAX/2.d0
       IF(C.GT.0.d0)GO TO 100
C --------------------------------------------------------------
--------------------
C FIND SUM OF Q50
C --------------------------------------------------------------
--------------------
       !DO 20 I=1,NUMBER
        DO 90 I=1,5555-1
           IF(A(I).LT.F)GO TO 90
           QSUM=QSUM+A(I)
           ISUM=ISUM+1
90    CONTINUE
       X=ISUM
       QAVG=QSUM/X
       CALL TRAPV(QAVG,BB,Z,E1,E2,XL,XN,V,Hriv)
       C=V/(V+1.7d0)
C --------------------------------------------------------------
--------------------
       WRITE(NUT,909)QMAX,QAVG,QAVG,V,C
100   IF(V0.GT.0.d0)V=V0
           TT=XL/3600.d0/V
           DELT=TT*C
           EX=(.08333d0+.5d0*DELT)/(1.5d0*DELT)
           CSTAR=1.d0-(1.d0-C)**EX
           CIN=CSTAR
           COUT=1.d0-CSTAR
           X=DELT*12.d0
           NUM=INT(X)
           IF((NUMBER+NUM+1).LE.576)GO TO 180
               NUMBER=NUMBER-NUM-1
180      CONTINUE
       Y=NUM
       DA=X-Y
       DB=1.d0-DA
C --------------------------------------------------------------
--------------------
       WRITE(NUT,911)CSTAR
C --------------------------------------------------------------
--------------------
C ROUTING LOOP AND FIND NEW QMAX
C --------------------------------------------------------------
--------------------
       WRITE(NUT,908)NZ1,NA,NZ2,NA,NA,NZ2,NA
       QMAX=0.d0
       !NUMB1=NUMBER+NUM+1 !lw TEST 5.1.2023
        NUMB1=5555-1-NUM-1 !LW TEST 5.18.2023
```

```
       QOUT=0.d0
       QIN=A(1)
       DO 200 I=1,NUMB1
           II=NUM+I+1
           QOUT=QOUT*COUT+QIN*CIN
           B(II)=B(II)+DA*QOUT
           II=II-1
           B(II)=B(II)+DB*QOUT
           TIME=TIME+.083333d0
           !IF(TIME.LT.TIME1.OR.TIME.GT.TIME2)GO TO 200
           !WRITE(NUT,913)TIME,A(I),B(I)

!IF(TIME.GE.TIME1.AND.TIME.LE.TIME2)WRITE(NUT,913)TIME,A(I),
   !  C    B(I),CC(I)
           QIN=A(I+1)
200   CONTINUE
       B(5555)=NUMBER+NUM+1
C --------------------------------------------------------------
----------
C  CALCULATE CONVOLUTION HYDROGRAPH BY ADDING OUTFLOW FROM NODE
(B(I)) AND RO HYDROGRAPH (CC(I))
C --------------------------------------------------------------
----------
        NUMBER = 0
        TIME=0.D0
        DO 210 I=1,5555-1
            seep=0.d0
            Qconv=B(I)+CC(I)
            IF (Qconv.gt.0.d0) then
              CALL TRAPV(Qconv,BB,Z,E1,E2,XL,XN,V,Hriv)
              IF (satk.gt.0.) then
              Seep=satk*(XL/hc)*(Hriv-hg)
              end if
              S(I)=Seep
              IF (seep.lt.0.001d0) then
                  seep=0.d0
              end if
              DD(I)=Qconv-seep
            else
                DD(I)=0.d0
            end if
            IF (DD(I).LT.0.D0) THEN
                DD(I)=0.D0
            END IF
            DD(I)=DD(I)+SNO
            IF (DD(I).GT.0) THEN
               NUMBER = NUMBER + 1
```

```
            END IF
            TIME=TIME+.083333d0

WRITE(NUT,913)TIME,A(I),B(I),CC(I),Qconv,S(I),SNO,DD(I)
210     CONTINUE
        DD(5555)=NUMBER
C --------------------------------------------------------------
----------
C  WRITE OUTFLOW HYDROGRAPHS
C --------------------------------------------------------------
----------
          !WRITE(NUT,913)TIME,A(I),B(I),CC(I),DD(I)
C --------------------------------------------------------------
----------
C  REASSIGN OUTFLOW TO MATRIX A AND WRITE TO STREAM AND NODE
MATRIX
C  SUM OUTFLOWS AND INFLOWS TO NODE
C --------------------------------------------------------------
----------
290   CONTINUE
      NUMBER = 0
       DO 300 I=1,5555-1
           A(I)=DD(I)
           SUMA24(1)=0.5D0*A(1)*5.D0*60.D0*0.0283168D0
           SEEP24(1)=0.5D0*S(1)*5.D0*60.D0*0.0283168D0
           IF (I.GT.1) THEN
           SUMA24(I)=SUMA24(I-1)+0.5d0*(A(I-1)+
     &        A(I))*5.d0*60.d0*0.0283168d0
           SEEP24(I)=SEEP24(I-1)+0.5d0*(S(I-1)+
     &        S(I))*5.d0*60.d0*0.0283168d0
           END IF
300     CONTINUE
          A(5555)=DD(5555)

C --------------------------------------------------------------
----------
C  CALCULATE BASEFLOW RECESSION
C --------------------------------------------------------------
----------
      IF (JDAY.GT.1) THEN
       IF (SUMA24(288).GT.(DAYQO(JDAY-1,INODE))) THEN
            dailyq=SUMA24(288)
            L=1
            DO WHILE (dailyq.gt.0.d0.and.L.le.213)
                 dqdt=arec*dailyq**brec
                 IF (dqdt.ge.dailyq) then
                      dqdt=dailyq
```

```
                 END IF
                 DBASEF(JDAY+L,INODE)=dailyq-dqdt +
DBASEF(JDAY+L,INODE)
                IF (DBASEF(JDAY+L,INODE).LT.0.D0) THEN
                    DBASEF(JDAY+L,INODE)=0.D0
               END IF
                 dailyq=dailyq-dqdt
                 L=L+1
            END DO
          END IF
      END IF

C --------------------------------------------------------------
----------
C  ASSIGN OTHER DAILY VALUES
C --------------------------------------------------------------
----------
        !DBASEF(JDAY+1,INODE)=DBASEF(JDAY+1,INODE)-
DBASEF(JDAY+1,JNODE)
        DBASEF(JDAY,INODE)=DBASEF(JDAY,INODE)-DBASEF(JDAY,JNODE)
        !IF (DBASEF(JDAY,INODE).LE.0.D0) THEN
        !  DBASEF(JDAY,INODE)=0.D0
        !END IF
        DAYQO(JDAY,INODE)=SUMA24(288)
        DSEEP(JDAY,INODE)=SEEP24(288)
        DAYDS(JDAY,INODE)=SUMQI24(288)-SUMA24(288)-SEEP24(288)
!INFLOW-OUTFLOW-SEEPAGE - accounts for flow still in-stream
        DSPRING(JDAY,INODE)=DBF(JDAY+1,JNODE)*3600.D0*24.D0
        DSNOW(JDAY,INODE)=DSNO(JDAY+1,INODE)*3600.D0*24.D0

      CALL
bfcalc(SISTORE,DBASEF,DRECH,DSM2,DTHETA2,INODE,dstorvol,
     & Dmaxstor,drloss,dSIWater1,dminstor)

       CALL MWRITE(NA,A)
      !ELSE
      !    WRITE(nut,999)
      !END IF
C --------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C --------------------------------------------------------------
----------
901   FORMAT(/,10X,'MODEL CHANNEL ROUTING BY CONVEX METHOD
WHERE',/,
     C 10X, 'A MODIFIED C-ROUTING COEFFICIENT IS ESTIMATED IN
ORDER',/,
```

```
     C 10X, 'TO ROUT THE STREAM',I2,' INFLOW HYDROGRAPH BY 5-
MINUTE',/,
     C 10X, 'INTERVALS (reference: National Engineering
Handbook,',/,
     C 10X, 'Hydrology, Chapter 17, page 17-52, August,1972,',/,
     C 10X, 'U.S. Department of Commerce).',/)

903   FORMAT(10X,'USER-SPECIFIED CHANNEL ROUTING COEFFICIENT =
',
     C F6.3,/)

904   FORMAT(10X,'USER-SPECIFIED CHANNEL AVG VELOCITY(FPS) = ',
     C F7.3,/)

905   FORMAT(10X,'ASSUMED REGULAR CHANNEL INFORMATION:',/,
     C  17X,'BASEWIDTH(FT) = ',F17.2,/,
     C  17X,'CHANNEL Z = ',F21.2,/,
     C  17X,'UPSTREAM ELEVATION = ',F12.2,/,
     C  17X,'DOWNSTREAM ELEVATION = ',F10.2,/,
     C  17X,'CHANNEL LENGTH(FT) = ',F12.2,/,
     C  17X,'MANNINGS FACTOR= ',F16.3,/,
     C  17X,'HYDRAULIC CONDUCTIVITY OF STREAMBED, FT/S
=',F16.8,/,
     C  17X,'THICKNESS OF STREAMBED CLOGGING LAYER, FT
=',F16.3,/,
     C  17X,'GROUNDWATER HEAD, FT =',F16.3,/)

909   FORMAT(11X,'CHANNEL ROUTING COEFFICIENT ESTIMATED:',/,
     C 14X,'MAXIMUM INFLOW(CFS) = ',F37.2,/,
     C 14X,'AVERAGE FLOWRATE IN EXCESS OF 50% MAXIMUM INFLOW =
',F8.2,/,
     C 14X,'CHANNEL NORMAL VELOCITY FOR Q = ',F6.1,' CFS =
',F14.2,
     C ' FPS',/,14X,'ESTIMATED CHANNEL ROUTING COEFFICIENT =
',F19.3,/)

908   FORMAT(/,11X,'CONVEX METHOD CHANNEL ROUTING RESULTS:',//,
     C 11X,' MODEL',8X,'INFLOW',5X,'OUTFLOW',5X,'DIRECT RO',3X,
     & 'CONVOLUTION',2X,'SEEPAGE',4X,'SNOW/SPRING',2X,
     & 'FINAL HYDROGRAPH'/,
     C 11X,'TIME',6X,'(NODE',I4,')',3X,'(STREAM',I2,')',3X,
     &
'(NODE',I4,')',2X,'(STREAM',I2,')',3X,'(STREAM',I2,')',4X,
     &    '(NODE',I4,')',2X,'(STREAM',I2,')',/,
     C 11X,'
(HRS)',6X,'(CFS)',10X,'(CFS)'7X,'(CFS)',8X,'(CFS)',1X,'
     &      (CFS)',6X,'(CFS)',6X,'(CFS)')
```

```
911   FORMAT(10X,' MODIFIED CHANNEL ROUTING COEFFICIENT FOR 5-
MINUTE ',/
     C ,14X,'UNIT INTERVALS IS CSTAR = ',F33.3,/)

913   FORMAT(10X,F7.3,2X,6F12.1)

999   FORMAT(10X,'NO FLOW OCCURRED ON THIS DAY')

      RETURN
      END SUBROUTINE CONVEX


C baseflow calculations
C  ------------------------------------------------------------
-----------------
      SUBROUTINE bfcalc(SISTORE,DBASEF,DRECH,DSM2,DTHETA2,INODE,
     C dstorvol,dmaxstor,drloss,dSIWater1,dminstor)
C  ------------------------------------------------------------
-----------------
      IMPLICIT DOUBLE PRECISION (a-h,o-z)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      DIMENSION
dstorvol(100),dmaxstor(100),dminstor(100),drloss(100),
     & dSIWater1(100)
      dimension SISTORE(100)
      DIMENSION DBASEF(5555,100),DRECH(5555,100),DSM2(5555,100),
     & DTHETA2(5555,100)

      storvol=dstorvol(inode)
      maxstore=dmaxstor(inode)
      Rlossfrac=drloss(inode)
      SIWater1=dSIWater1(inode)
      minstore=dminstor(inode)
      DSWC2=0.D0

      IF (DBASEF(JDAY,INODE).GT.0.0D0) THEN
          BFLOSS=DBASEF(JDAY,INODE)
          RECHARGE1=0.D0
          DSWC2=SISTORE(INODE)-MINSTORE
          IF (BFLOSS.GT.DSWC2) THEN
              BFLOSS = DSWC2
              DBASEF(JDAY,INODE)=BFLOSS
              SISTORE(INODE)=MINSTORE
          ELSE
             SISTORE(INODE)=SISTORE(INODE)-BFLOSS
```

```
          END IF
      ELSE
          BFLOSS=0.D0
          RECHARGE1=-1.D0*DBASEF(JDAY,INODE) !MAKE IT POSITIVE
gROUNDWATER
          DBASEF(JDAY,INODE)=0.D0
          DSWC2=SISTORE(INODE)-MINSTORE
          !print*,'dswc2',dswc2
          IF (RECHARGE1.GT.DSWC2.and.DSWC2.GT.0) THEN
              RECHARGE1 = DSWC2
              SISTORE(INODE)=MINSTORE
          ELSE IF (DSWC2.LE.0.D0) THEN
              RECHARGE1 = 0
          ELSE
             SISTORE(INODE)=SISTORE(INODE)-RECHARGE1
          END IF
      END IF
c --- Recalculate shallow infiltration water volume by removing
baseflow losses

          !SISTORE(INODE)=SISTORE(INODE)-BFLOSS-RECHARGE1
C --- Recalculate soil water then calculate recharge accordingly
      If (SISTORE(INODE).GT.MAXSTORE) THEN
          BFLOSS2=SISTORE(INODE)-MAXSTORE
          DBASEF(JDAY,INODE)=DBASEF(JDAY,INODE)+BFLOSS2
          SISTORE(INODE)=MAXSTORE
          RECHARGE=Rlossfrac*SISTORE(INODE)
          SISTORE(INODE)=SISTORE(INODE)-RECHARGE
      !if (jday.eq.1) then
      !    print*,'storvol
etc',inode,storvol,maxstore,Rlossfrac,SIWater1
      !    print*,'swc2',inode,swc2,sistore(inode)
      !    end if
      !IF (SWC2.le.wp) then
      !      recharge=0.D0
      else
          RECHARGE=Rlossfrac*SISTORE(INODE)
          DSWC2=SISTORE(INODE)-MINSTORE
          IF (RECHARGE.GT.DSWC2.AND.DSWC2.GT.0) THEN
              RECHARGE=DSWC2
              SISTORE(INODE)=MINSTORE
          ELSE
              RECHARGE=0.D0
          END IF
          SISTORE(INODE)=SISTORE(INODE)-RECHARGE
      END IF
```

```
      RECHARGE= RECHARGE+RECHARGE1
      !pRINT*,' CONVEX RECHARGE,RECH1',RECHARGE,RECHARGE1
      DSIwater = SISTORE(INODE)-SIwater1
      SWC2=SISTORE(INODE)/storvol

      DRECH(JDAY,INODE)=RECHARGE + DRECH(JDAY,INODE) !ADDS
RECHARGE FROM UHCN/GASH DPSEEP PROCESS
      DSM2(JDAY,INODE)=DSIwater
      DTHETA2(JDAY,INODE)=SWC2
      !IF (JDAY.EQ.1) THEN
      !    PRINT*,'DSIWATER,RECHARGE',dsiWATER, dsm2(jday,inode)
      !end if

      RETURN
      END SUBROUTINE BFCALC
```

```
C PROGRAM MAIN - Marco Pazmi�o-Hernandez, Rafael Mu�oz-Carpena
C ------------------------------------------------------------------
-----------
      PROGRAM CUENCA
C ------------------------------------------------------------------
----------
C   MAIN DRIVER FOR CUENCA ROUTING PROGRAM
C ------------------------------------------------------------------
----------
C Modifications
C Inputs in Green-Ampt and UHCN processes standardized - LLW
10.06.2022
C      DEFINITIONS
C SS: STORAGE MATRIX FOR STREAMFLOW VALUES
C DPRECIP: PRECIPITATION MATRIX
C SNODE: STREAMFLOW STORAGE MATRIX BY NODE
C STAIL: STREAMFLOW STORAGE MATRIX FOR FLOWS OCCURING AFTER 24
HOURS BY NODE
C NUT: OUTPUT FILE (".OANS")
C NDAT: LINK AND NODE INPUT PARAMETERS (".IDAT")
C NIPR: PRECIPITATION BY NODE INPUTS (".IPRN")
C NSSS: STREAMFLOW MATRIX BY STREAM WRITTENT TO FILE (".OSSS")
C NZ2: DOWNSTREAM NODE OF EACH PROCESS WHERE FLOWS ARE
CALCULATED
C A(5555):
C SUM(100):
C SUM24(100):
C LISFIL: READS NAMES FOR INPUT AND OUTPUT FILES
C ----STORAGE MATRICES----
C DAYRO(5555,100): DAILY RUNOFF (MM)
C DAYQI(5555,100): DAILY STREAMFLOW AT EACH NODE (MM)
C DAYDN(5555,100): DAILY DRAINAGE AT EACH NODE (MM)
C DAYDS(5555,100): DAILY DETENTION STORAGE AT EACH NODE (MM)
C DAYRO(5555,100): DAILY RUNOFF AT EACH NODE (MM)
C DAYMO(5555,100): DAILY FLOW MOVED FROM EACH NODE (MM)
C FIRR(1,100): FRACTIONAL WATERSHED AREA OF IRRIGATION AT EACH
NODE (MM)
C ------------------------------------------------------------------
----------
C   DECLARE VARIABLES
C ------------------------------------------------------------------
----------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)
      COMMON/INPUTS/DATAINP(100,100)
```

```
COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE

COMMON/NNINOUT/NIET,NIBF,NITM,NITU,NITL,NIWS,NISR,NICK,NIAB,NIIR
,
     &
NOPR,NORO,NOET,NODN,NOSM,NOQI,NODS,NOAB,NOIR,NOQO,NOMO,NOPE,NOWB
,
     & NOSD,NISN,NOIN
      DIMENSION
dstorvol(100),dmaxstor(100),dminstor(100),drloss(100),
     & dSIWater1(100)
      DIMENSION A(5555),SUMD(100),SUMD24(100)
      DIMENSION DAYRO(5555,100),DAYQI(5555,100),DAYDN(5555,100),
     &
DAYDS(5555,100),DAYMO(5555,100),DAYQO(5555,100),DSEEP(5555,100),
     & DTHETA(5555,100),DETA(5555,100),DSED(5555,100),
     &
DRECH(5555,100),DSM1(5555,100),DSM2(5555,100),DBASEF(5555,100),
     & DSPRING(5555,100),DSNOW(5555,100),DTHETA2(5555,100)
      DIMENSION
FIRR(1,100),PEFF(5555,100),WBAL(5555,100),DAREA(100)
      DIMENSION dstore(100),vstore(100),Pstore(100),sstore(100),
     & SISTORE(100),dkode(100),DIRREFF(5555,100)
      CHARACTER*75 LISFIL(31)

common/CINPUT/DETO(5555,100),DBF(5555,100),DTAVG(5555,100),
     &
DTMAX(5555,100),DTMIN(5555,100),DWS2(5555,100),DSORAD(5555,100),
     &
DCKM(5555,100),DAB(5555,100),DIRR(5555,100),DSNO(5555,100)
      DIMENSION
dinflow(5555,100),doutflow(5555,100),deltast(5555,100)

C-----------------------------------------------------------------
----------
C  INITIALIZE AND ADD NODES TO FIRST ROW OF NODE HYDROGRAPH
MATRIX
C-----------------------------------------------------------------
---------
      DO 15 I=1,5555
          DO 15 K=1,100
              SNODE(I,K)=0.D0
              STAIL(I,K)=0.D0
              DPRECIP(I,K)=0.D0
              DETO(I,K)=0.D0
              DTAVG(I,K)=0.D0
```

```
              DTMAX(I,K)=0.D0
              DTMIN(I,K)=0.D0
              DWS2(I,K)=0.D0
              DSORAD(I,K)=0.D0
              DCKM(I,K)=0.D0
              DBF(I,K)=0.D0
15    CONTINUE
C -------INITIALIZE OUTPUT MATRICES--------------
      DO 17 I=1,5555
          DO 17 K=1,100
              DAYRO(I,K)=0.D0 ! Stores daily runoff values (mm)
              DAYQI(I,K)=0.D0 ! Stores daily streamflow values
(mm)
              DAYQO(I,K)=0.D0 ! Stores daily outflow values (mm)
              DAYDN(I,K)=0.D0 ! Stores daily drainage
(infiltration) losses (mm)
              DAYDS(I,K)=0.D0 ! Stores daily storage values (mm)
              DAYMO(I,K)=0.D0 ! Stores daily permanently moved
flows (mm) (i.e. if a stream is split and the flows leave the
watershed) --
              PEFF(I,K)=0.D0  ! Stores daily effective
precipitation values
              WBAL(I,K)=0.D0  ! Stores daily water balance sum
by node
              DETA(I,K)=0.D0  ! Stores daily actual ET values
(mm)
              DTHETA(I,K)=0.D0! Stores daily end-of-day soil
water content values (mm)
              DSED(I,K)=0.D0  ! Stores daily sediment
concentration (g/L)
              DRECH(I,K)=0.D0 ! Stores dailly recharge values
(m3)
              DSM1(I,K)=0.D0  ! stores daily change in soil root
zone moisture storage (m3)
              DSM2(I,K)=0.D0  ! Stores daily change in soil
intermediate water storage (m3)
              DBASEF(I,K)=0.D0! Stores daily baseflow
contributions (m3)
              DSNO(I,K)=0.D0  ! Stores daily snowmelt
contributions (m3/s)
              DSEEP(I,K)=0.D0 ! Stores daily seepage from convex
process (m3)
              DSPRING(I,K)=0.D0 ! Stores daily spring
contributions (DBF converted to (m3))
              DSNOW(I,K)=0.D0 ! Stores daily snowmelt
contributions (DSNO converted to (m3)
              DINFLOW(I,K)=0.D0
```

```
              DOUTFLOW(I,K)=0.D0
              DTHETA2(I,K)=0.D0
              DIRREFF(I,K)=0.D0
17    CONTINUE
c -------Initialize daily initial conditions arrays -----------
      DO 20 K=1,100
          PSTORE(K)=0.D0 !Pipe storage
          SSTORE(K)=0.D0 !Flowby dead storage
          DSTORE(K)=0.D0 !Flowthru dead storage
          VSTORE(K)=0.D0 !Flowthrough effective volume storage
          SISTORE(K)=0.D0 !Deep percolation storage
          DKODE(K)=0.D0
          DSTORVOL(K)=0.D0
          DRLOSS(K)=0.D0
          DMAXSTOR(K)=0.D0
          DSIWATER1(K)=0.D0
20    CONTINUE


C ----------------------------------------------------------------
---------
C  INITIAL FILES
C ----------------------------------------------------------------
---------
      NDAT=5    ! (file=".idat")
      NUT=6     ! (file=".oans")
      NSSS=7    ! (file=".osss")
      NIPR=8    ! (file=".iprn") Precipitation
      NDSS=9    ! (file=".odss") Additive streamflow at each
node (m^3)
      NIET=10   ! (file=".ieto") Evapotranspiration
      NIBF=11   ! (file=".ibfl") Baseflow
      NITM=12   ! (file=".itmp") Average daily temperature (C)
      NITU=13   ! (file=".itma") Maximum daily temperature (C)
      NITL=14   ! (file=".itmi") Minimum daily temperature (C)
      NIWS=15   ! (file=".iwsp") Average daily wind speed (cm/s)
      NISR=16   ! (file=".isor") Average daily solar radiation
(Langley/day)
      NICK=17   ! (file=".ickm") Mid-season crop coefficient
(maximum), Kc,mid (-). The value can be varied daily with a
phenological curve during the season for the specific plant.
Free format
      NIAB=18   ! (file=".iabs") water abstractions
      NIIR=19   ! (file=".iirr") irrigation
      NOPR=20   !!! (file=".oprn")*currently equal to
corresponding input
      NORO=21   !!! (file=".odro") Runoff (mm)
```

```
      NOET=22   !!! (file=".oeta") calculated from ThetaFAO
subroutine
      NODN=23   !!! (file=".odng") daily drainage at each node
(mm)
      NOSM=24   ! (file=".osmi") calculated from ThetaFAO
subroutine
      NOQI=25   !!! (file=".oqif") Stream inflow at each node
(mm)
      NODS=26   !!! (file=".odst") Storage in detention
structures (mm)
      NOAB=27   ! (file=".oabs")*currently equal to
corresponding input
      NOIR=28   ! (file=".oirr")*currently equal to
corresponding input
      NOQO=29   !!! (file=".oqof") Stream outflow at each node
(mm)
      NOMO=30   !!! (file=".odmo") Stream flow moved away from
node (mm)
      NOPE=31   !!! (file=".opef") Daily effective precipitation
(mm)
      NOWB=32   ! (file=".owbl") Daily water balance sum (mm)
      NOSD=33   ! (file=".osed") Daily sediment concentration
(g/L)
      NISN=34   ! (file=".isno") Daily snowmelt contributions
from upstream (!!!need to decide units and if this will be read
in as a lump sum or divided over the whole day)
      NOIN=35   ! (fil.=".oinp") Reads and prints input matrices
directly
C ---------------------------------------------------------------
---------
C  Call input file to process and create output files of same
name
C ---------------------------------------------------------------
---------
      call finput(LISFIL)
C ---------------------------------------------------------------
----------
      WRITE(NUT,701)
      WRITE(NUT,702)
      WRITE(NUT,701)
C ---------------------------------------------------------------
----------
C  PROCESS INPUT DATA FILE !Subroutines CUENCA
C ---------------------------------------------------------------
----------
      READ(NIPR,*)NRDAYS,NRNODES,ISUBDAYOUT !ISUBDAYOUT = 0 DO
NOT INCLUDE SUBDAILY OUTPUT, 1 = INCLUDE SUBDAILY OUTPUT
```

```
      READ(NIPR,*) !SKIP LINE

C---------------------------------------------------------------
---------
C  LOOP TO READ INPUT FILE AS MATRIX LOOP
C---------------------------------------------------------------
---------
      CALL READDATA(NRNODES,NRDAYS,FIRR,DAREA)
      !print*,'snow',dsno(1,1)
C---------------------------------------------------------------
---------
C  START DAILY TIME LOOP
C---------------------------------------------------------------
---------
          write(*,*)'*** Running simulation days '
          DO 100 JDAY=1,NRDAYS
              write(stdout,'(i4)',advance='no')jday
                 flush(stdout)
              DO 1216 I=1,5555
                  DO 1216 K=1,10
                      SS(I,K)=0.D0  !Stores flows by stream
number
1216          CONTINUE
              DO 1220 I=1,5555
                  DO 1218 K=1,100
                      SNODE(I,K)=0.D0     !Stores flows by node
number
1218              CONTINUE
1220          CONTINUE
C --------------------------------------------------------------
---------
C  READ STARTING NODE NUMBER
C --------------------------------------------------------------
---------
              NZ2=INT(DPRECIP(1,1))
C --------------------------------------------------------------
---------
C  START LINK AND NODE LOOP
C --------------------------------------------------------------
---------
              WRITE(NUT,804)JDAY
              WRITE(NSSS,905)JDAY
              JCOUNT=1
              KODE=0
              DO WHILE (KODE.NE.999)
                  NZ1=INT(DATAINP(JCOUNT,1)) !Initial NZ1
                  NZ2=INT(DATAINP(JCOUNT,2)) !Initial NZ2
```

```
                  KODE=INT(DATAINP(JCOUNT,3)) !Initial KODE
                  !print*,'jday,n1,n2,proc',jday,nz1,nz2,kode
                  DO 21 J=1,100
                      IF (INT(DPRECIP(1,J)).EQ.NZ2) THEN
                       INODE=J
                      END IF
                      IF (INT(DPRECIP(1,J)).EQ.NZ1) THEN
                      JNODE=J
                      END IF
21                CONTINUE
                  IF (KODE.LT.13) THEN
                  DKODE(INODE)=KODE
                  END IF
                  IF (KODE.NE.999) THEN
                      WRITE(NUT,601)
                      WRITE(NUT,600)NZ1, NZ2, KODE
                      WRITE(NUT,601)
                      CALL
PCALC(P,FIRR,DIRREFF,DAYQO,JNODE,DAREA,PEFF)
                      CALL
PROCESSES(P,DSTORE,VSTORE,DBASEF,PEFF,
     &
DAYRO,DAYQI,DAYQO,DAYDN,DAYDS,DAYMO,DTHETA,DETA,
     &
PSTORE,SSTORE,SISTORE,DSED,DRECH,DSM1,DSM2,DSEEP,
     &
DSPRING,DSNOW,DTHETA2,DSTORVOL,DMAXSTOR,DRLOSS,
     &               dSIWater1,dminstor,DBF,DIRREFF)
                      WRITE(NUT,701)
                  ELSE
                      WRITE(NUT,701)
                      WRITE(NUT,904)JDAY
C -------------------------------------------------------------
----------------
C  WRITE RESULTS TO NSSS - TEMPORAL FILE .OSSS
C -------------------------------------------------------------
----------------
                      WRITE(NSSS,701)   !writes file formatting
(======) at top

WRITE(NSSS,903)(INT(DPRECIP(1,K)),K=1,NRNODES) !writes the node
headers
                      DO 1205 K=1,NRNODES !INITIALIZE SUMD(K)
AND SUMD24(K)
                          sumd(K)=0.d0
                          sumD24(K)=0.d0
1205                  CONTINUE
```

```
C ---- write time and snode matrix into .osss output file
                      DO 1200 I=1,5555

WRITE(NSSS,910)I*5.d0/60.d0,(SNODE(I,K),
     &                         K=1,NRNODES)
                          IF (JDAY.EQ.2.AND.K.EQ.10) THEN
                              PRINT*,'STAIL',I,K,STAIL(I,K)
                          END IF
1200                  CONTINUE
c ----Calculate daily sum and event sum
                      !DO 1201 K=1,NRNODES
                      !    DO 1201 L=1,5555-1
                      !        STAIL(L,K)=0.D0
1201                  CONTINUE
                      DO 1210 K=1,NRNODES
                          SUMD(K)=0.5D0*snode(1,k)*5.D0*60.D0*
     &                     0.0283168D0
                          SUMD24(K)=0.5D0*snode(1,k)*5.D0*60.D0*
     &                     0.0283168D0
                          DO 1210 L=2,5555-1
                              sumD(K)=sumD(K)+0.5d0*(SNODE(L-
1,K)+
     &
SNODE(L,K))*5.d0*60.d0*0.0283168d0 !Calculating the sum by
taking average of timesteps, multiplying over time step to get
volume, and converting to cubic meters
                              IF(L.LE.288) THEN   !if Time of
flows is less than or equal to 24 hours, do this LW 4.2.2022
                                  sumD24(K)=sumD24(K)+0.5d0*
     &                            (SNODE(L-1,K)+SNODE(L,K))*
     &                            5.d0*60.d0*0.0283168d0
                              End if
                              !    stail(l-288,k)=snode(l,k)
!storing flows after 24 hours to the stail matrix lw 4.2.2022
                              !end if
                              !if (snode(5555,k).ge.288) then
                              !    stail(5555,k)=snode(5555,k)-
288.d0
                              !else
                              !    stail(5555,k)=0.d0
                              !end if
1210                      CONTINUE
!                      DO 1215 K=1,NRNODES
!                          DO 1215 L=1,5555
!                              SNODE(L,K)=STAIL(L,K)
!1215                  CONTINUE
C---- Write summary at end of each day
```

```
                      WRITE(NSSS,701)
                      write(NSSS,909)'24H RAIN',(PEFF(JDAY,K),
!EDITED 6.8.2023 LW
     &                 K=1,NRNODES)
                      write(NSSS,909)'24H
VOL',(SUMD24(K),K=1,NRNODES)
                      write(NSSS,909)'TOT
VOL',(SUMD(K),K=1,NRNODES)
                      write(NSSS,909)'>24H VOL',
     &                (SUMD(K)-SUMD24(K),K=1,NRNODES)
                      WRITE(NSSS,701)
C---- write to daily summary file

write(NDSS,911)JDAY,(SUMD24(K),K=1,NRNODES)
C---------------------------------------------------------------
----------------
C  INCREASE JCOUNT TO MOVE TO NEXT PROCESS, ZERO OUT STORAGE
MATRICES
C---------------------------------------------------------------
----------------
                  END IF
                  JCOUNT=JCOUNT+1

              END DO
100       CONTINUE
C --------------------------------------------------------------
-----------------
C DAILY WATER BALANCE CALCULATION
C --------------------------------------------------------------
-----------------
          DO 101 L=1,NRDAYS
          DO 102 K=1,NRNODES
c              WBAL(L,K)=PEFF(L,K)+DAYQI(L,K)-DAYRO(L,K)!-
DAYDS(L,K)
c     &         -DAYDN(L,K)-DAYMO(L,K)-DAYQO(L,K)-DETA(L,K)
              !WBAL(L,K)=PEFF(L,K)-DAYRO(L,K)-DAYDN(L,K)
    ! &         -DAYDS(L,K)-DAYMO(L,K)-DAYQO(L,K)-DETA(L,K)
102       CONTINUE
101   CONTINUE
      WRITE(NOWB,920)
      WRITE(NOWB,921)
      DO 105 L=1,NRDAYS
          DO 106 K=1,NRNODES
              DINFLOW(L,K)=PEFF(L,K)+DAYQI(L,K)
!              DOUTFLOW(L,K)=DAYQO(L,K)+DETA(L,K)+
!     &                    DAYMO(L,K) +DRECH(L,K)
```

```
DOUTFLOW(L,K)=DAYQO(L,K)+DETA(L,K)+DAYRO(L,K)+!DAYDN(L,K)+
     &        DAYMO(L,K)
+DRECH(L,K)+DAYDS(L,K)+DSEEP(L,K)+DBASEF(L,K)+
     &        DBF(L,K)
              DELTAST(L,K)=DSM1(L,K)+DSM2(L,K) ! DSM1 and DSM2
are negative when there is a loss, so sign must be changed
              WBAL(L,K)=DINFLOW(L,K)-DOUTFLOW(L,K)-DELTAST(L,K)
              IF (DINFLOW(L,K).EQ.0.D0) THEN
                  PERDIFF=0.D0
              ELSE
                  PERDIFF=(WBAL(L,K)/DINFLOW(L,K))*100.D0
                  IF (PERDIFF.GT.999.D0) THEN
                      PERDIFF=999.D0
                  END IF
              END IF

WRITE(NOWB,912)L,INT(DPRECIP(1,K)),INT(DKODE(K)),PEFF(L,K)
     C
,DSPRING(L,K),DSNOW(L,K),DAYQI(L,K),DAYQO(L,K),
     C
DAYRO(L,K),DETA(L,K),DSEEP(L,K),DBF(L,K),DBASEF(L,K),
     C
DRECH(L,K),DAYMO(L,K),DAYDS(L,K),DAYDN(L,K),DSM1(L,K),
     C            DSM2(L,K),WBAL(L,K),PERDIFF
     C         !DSM2(L,K),0.d0,0.d0
             !IF(L.EQ.16.AND.K.EQ.6) THEN
             !    print*,wbal(16,6)
             !    print*,DInflow(16,6)
             !    print*,dayqi(16,6)
             !END IF
c              WBAL(L,K)=PEFF(L,K)+DAYQI(L,K)-DAYRO(L,K)!-
DAYDS(L,K)
c     &         -DAYDN(L,K)-DAYMO(L,K)-DAYQO(L,K)-DETA(L,K)
              !WBAL(L,K)=PEFF(L,K)-DAYRO(L,K)-DAYDN(L,K)
    ! &         -DAYDS(L,K)-DAYMO(L,K)-DAYQO(L,K)-DETA(L,K)
106       CONTINUE
105   CONTINUE

c testing testing testing
!      DO 107 K=1,NRNODES
!          DO 108 L=1,NRDAYS
!              print*,'day, wbal,dsed',l,wbal(l,k),dsed(l,k)
!108       CONTINUE
!107   CONTINUE
```

```
C ---------------------------------------------------------------
-----------------
C   WRITE TO DAILY OUTPUT FILES
C ---------------------------------------------------------------
-----------------
      DO 110 J=1,NRDAYS
          WRITE(NORO,911)J,(DAYRO(J,K),K=1,NRNODES)
          WRITE(NODN,911)J,(DAYDN(J,K),K=1,NRNODES)
          WRITE(NOQI,911)J,(DAYQI(J,K),K=1,NRNODES)
          WRITE(NODS,911)J,(DAYDS(J,K),K=1,NRNODES)
          WRITE(NOQO,911)J,(DAYQO(J,K),K=1,NRNODES)
          WRITE(NOMO,911)J,(DAYMO(J,K),K=1,NRNODES)
          WRITE(NOPE,911)J,(PEFF(J,K),K=1,NRNODES)
          WRITE(NOPE,911)J,(DIRREFF(J,K),K=1,NRNODES)
          WRITE(NOET,911)J,(DETA(J,K),K=1,NRNODES)
          WRITE(NOSM,911)J,(DTHETA(J,K),K=1,NRNODES)
          !WRITE(NOWB,912)J,(WBAL(J,K),K=1,NRNODES)
          WRITE(NOSD,911)J,(DSED(J,K),K=1,NRNODES)
110   CONTINUE
      DO 120 J=1,NRDAYS
          WRITE(NOSM,911)J,(DTHETA2(J,K),K=1,NRNODES)
120   CONTINUE
      Write(NOSD,907)
C ---------------------------------------------------------------
----------------
      CLOSE(NDAT) ! close ".idat" file
      CLOSE(NUT)  ! close ".oans"  file
      CLOSE(NIPR)  ! close "iprn"  file
      CLOSE(NSSS)  ! close ".osss"  file
      CLOSE(NDSS)
      CLOSE(NIET)
      CLOSE(NIBF)
      CLOSE(NIAB)
      CLOSE(NIIR)
      CLOSE(NOPR)
      CLOSE(NORO)
      CLOSE(NOET)
      CLOSE(NODN)
      CLOSE(NOSM)
      CLOSE(NOQI)
      CLOSE(NODS)
      CLOSE(NOAB)
      CLOSE(NOIR)
      CLOSE(NOQO)
      CLOSE(NOMO)
      CLOSE(NOPE)
      CLOSE(NOWB)
```

```
      CLOSE(NOSD)
      CLOSE(NOIN)

C -------------------------------------------------------------
-----------------
C    OUTPUT - FORMAT
C -------------------------------------------------------------
-----------------
600   FORMAT(3X,'FLOW PROCESS FROM NODE ',I5,' TO NODE ',I5,
     C' IS CODE = ',I3)
601   FORMAT(1X,76('*'))
602   FORMAT(1X,'*** FATAL READING ERROR - CHECK DATA INPUT
***')
701   FORMAT(1X,120('='))
702   FORMAT(19X,'C U E N C A   R O U T I N G   A N A L Y S I
S')
804   FORMAT(17X,'>>> START OF  CUENCA  ROUTING  ANALYSIS DAY
',I4,
     C ' <<<')
903   FORMAT(3X,'TIME(h)',100(7X,'(',I3,')'))
904   FORMAT(17X,'>>> END  OF  CUENCA  ROUTING  ANALYSIS DAY
',I4,
     C ' <<<')
C905   FORMAT(20X,'>>>>> SS - FILE TEMPORAL DATA <<<<<',/,
!print streams instead of nodes
c     C    1X,76('='),/,
c     C
9X,'SS',10X,'SS',10X,'SS',10X,'SS',10X,'SS',10X,'SS',/,
c     C
8X,'(1)',10X,'(2)',9X,'(3)',9X,'(4)',9X,'(5)',9X,'(6)')
905   FORMAT(17X,'>>>> SNODE (CFS)- FILE TEMPORAL DATA DAY'
,I2,' <<<<')          !print nodes instead of streams
c906   FORMAT(1X,6F12.3)
906   FORMAT(I4,6E12.3)
c703   FORMAT(1X,76(':'))
907   FORMAT(10X,'>>> CUENCA: FINISHED NORMAL EXECUTION <<<',/)
908   FORMAT(100E12.3)
909   FORMAT(A12,101E12.5)
910   FORMAT(F12.3,100E12.5)
911   FORMAT(I5,3X,100E12.5)
912   FORMAT(3I6,3X,16E12.4,2f12.2)
920   FORMAT(10x,"INPUTS",18X,"OUTPUTS",85X,"CHANGE IN SOIL
WATER
     C STORAGE")
921
FORMAT(3X,'DAY',3X,'NODE',3x,'PROC.',3X,'PRECIP',5X,'Springflow'
,
```

```
     C 3x,'Snowmelt',4x,'SWI',9X,'SWO',9x,'RO',10x,'ET',10x,
     C
'Seepage',5x,'Basef(TF)',3x,'Baseflow',4x,'GWR',9x,'MOVE',8X,
     C
'DETENTION',3X,'INFIL.',6X,'THETA1',6X,'THETA2',6X,'ERROR',
     C 3X,'PERCENTERROR')
      STOP 'FINISHED!'
      END PROGRAM CUENCA

C  -------------------------------------------------------------
---------------
C   SUBROUTINE PROCESSES - ! PROGRAM 15  - Based CUENCA
C  -------------------------------------------------------------
---------------
      SUBROUTINE PROCESSES(P,DSTORE,VSTORE,DBASEF,PEFF,
     &
DAYRO,DAYQI,DAYQO,DAYDN,DAYDS,DAYMO,DTHETA,DETA,
     &
PSTORE,SSTORE,SISTORE,DSED,DRECH,DSM1,DSM2,DSEEP,
     &                DSPRING,DSNOW
,DTHETA2,DSTORVOL,DMAXSTOR,DRLOSS,
     &                dSIWater1,dminstor,DBF,DIRREFF)
C --------------------------------------------------------------
----------
C   DECLARE VARIABLES
C --------------------------------------------------------------
----------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      DIMENSION DAYRO(5555,100),DAYQI(5555,100),DAYDN(5555,100),
     &
DAYDS(5555,100),DAYMO(5555,100),DAYQO(5555,100),PEFF(5555,100),
     &
DTHETA(5555,100),DETA(5555,100),DSED(5555,100),DSEEP(5555,100),
     &
DRECH(5555,100),DSM1(5555,100),DSM2(5555,100),DBASEF(5555,100),
     & DSPRING(5555,100),DSNOW(5555,100),DTHETA2(5555,100)
      DIMENSION dstore(100),vstore(100),Pstore(100),sstore(100),
     & SISTORE(100)
      DIMENSION
dstorvol(100),dmaxstor(100),dminstor(100),drloss(100),
     & dSIWater1(100),DBF(5555,100),DIRREFF(5555,100)
C  -------------------------------------------------------------
---------------
C   INITIALIZE CUENCA KODE PROCESSES
```

```
C  -------------------------------------------------------------
---------------
c      IF(KODE.EQ.1)CALL unith(m,n,m1,n1,NZ1,KODE,mn1,mn2)
      IF(KODE.EQ.1)STOP "UNITH is not available for continuous
     & simulation"
      IF(KODE.EQ.2)CALL flowby(DAYQI,DAYQO,DAYMO,DAYDS,
     & SSTORE)
      IF(KODE.EQ.3)CALL fthru(DAYDS,DAYQI,DAYQO,DSTORE,
     & VSTORE)
      IF(KODE.EQ.4)CALL piper(DAYQI,DAYQO,DAYDS,PSTORE)
      IF(KODE.EQ.5)CALL
convex(DAYQI,DAYQO,DAYDN,DAYDS,DSEEP,DSPRING,
     &
DSNOW,DBASEF,DRECH,DSM2,SISTORE,DTHETA2,dstorvol,dmaxstor,drloss
,
     & dSIWater1,dminstor,dirreff)
      IF(KODE.EQ.6)CALL clear(DAYQI,DAYQO,DAYMO)
      IF(KODE.EQ.7)CALL add(DAYQI,DAYQO)
      IF(KODE.EQ.8)CALL split(DAYQI,DAYQO,DAYMO)
      IF(KODE.EQ.9)CALL move(DAYQI,DAYQO,DAYDS)
      IF(KODE.EQ.10)CALL hydrog(DAYQI,DAYQO)
      IF(KODE.EQ.11)CALL
uhcn(P,DAYRO,DAYDN,DSM1,DSM2,DRECH,DBASEF,
     & DAYMO,DTHETA,DETA,SISTORE,BFloss,DSED,PEFF,DTHETA2,
     &  dstorvol,dmaxstor,drloss,dSIWater1,dminstor,DBF,DIRREFF)
      IF(KODE.EQ.12)CALL
gash(P,DAYRO,DAYDN,DSM1,DSM2,DRECH,DBASEF,
     & DAYMO,DTHETA,DETA,SISTORE,BFloss,DSED,PEFF,DTHETA2,
     &  dstorvol,dmaxstor,drloss,dSIWater1,dminstor,DBF,DIRREFF)
      IF(KODE.EQ.13)CALL prnode
      IF(KODE.GT.13) WRITE(NUT,602)
C --------------------------------------------------------------
-----------------
C    OUTPUT - FORMAT
C --------------------------------------------------------------
-----------------
c701   FORMAT(1X,76('='))
c702   FORMAT(19X,'C U E N C A   R O U T I N G   A N A L Y S I
S')
c703   FORMAT(1X,76(':'))
602   FORMAT(1X,'*** FATAL READING ERROR (KODE>12) - CHECK INPUT
***')

      RETURN
      END SUBROUTINE PROCESSES
```

```
      C ---------------------------------------------------------------
------------------------------------
C   Program:
C ---------------------------------------------------------------
------------------------------
      SUBROUTINE
dpseep(ISOIL,SISTORE,DPerc,INODE,BFloss,NA,AREA,
     C
soilpt,Zstore,RECHARGE,DSIwater,FC,WP,wcini,SWC2,
     C
DBASEF,DRECH,DSM2,dstorvol,dmaxstor,drloss,
     C                dSIWater1,dminstor,DBF,DTHETA2)
C ---------------------------------------------------------------
------------------------------
C To calculate soil moisture redistribution over time based
C Adapted from ACRU 3.0 Hydrological Modelling system, R.E.
Schulze, 1995
C ---------------------------------------------------------------
------------------------------
      implicit double precision (a-h, o-z)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      dimension
sfrac1(21),sfrac2(21),SISTORE(100),A(5555),subp(21)
      DIMENSION DBASEF(5555,100),DRECH(5555,100),DSM2(5555,100)
      DIMENSION
dstorvol(100),dmaxstor(100),dminstor(100),drloss(100),
     & dSIWater1(100),DBF(5555,100),DTHETA2(5555,100)
C ---------------------------------------------------------------
------------------------------
C SIWater: water stored as soil water (i.e. shallow
infiltration) available for loss to deep percolation
C BFlossfrac: Fraction of water lost daily to stream baseflow
from SIWater
C BFloss: Actual volume (depth) of water lost daily to
streamflow
C SIWater2: Shallow infiltration water available after losses to
baseflow
C Recharge: Fraction of water lost daily to aquifer recharge
C Rlossfrac: Fraction of water lost daily to aquifer recharge
C Dperc: water added to shallow infiltration water available
from ThetaFAO program (mm)
C ---Passed variables---
C Area: area in ha
C ---------------------------------------------------------------
----------
```

```
c! DATA typesoil/'Clay','Silty clay','Sandy clay','Silty clay
loam',
c!C  'Clay loam','Sandy clay loam','Silt','Silt loam','Loam',
c!C  'Very fine sandy loam','Fine sandy loam','Sandy loam',
c!C  'Coarse sandy loam','Loamy very fine sand','Loamy fine
sand',
c!C  'Loamy sand','Loamy coarse sand','Very fine sand',
c!C  'Fine sand','Sand','Coarse sand'/
      DATA
sfrac1/0.25d0,0.35d0,.40d0,0.35d0,0.40d0,0.50d0,0.45d0, !
fraction lost to baseflow adapted from ACRU
     C            0.45d0,0.5d0,0.65d0,0.65d0,0.65d0,0.65d0,
     C            0.70d0,0.70d0,0.70d0,0.70d0,0.80d0,0.80d0,
     C            0.80d0,0.80d0/
      DATA
sfrac2/0.25d0,0.35d0,.40d0,0.35d0,0.40d0,0.50d0,0.45d0, !
fraction lost to recharge
     C            0.45d0,0.5d0,0.65d0,0.65d0,0.65d0,0.65d0,
     C            0.70d0,0.70d0,0.70d0,0.70d0,0.80d0,0.80d0,
     C            0.80d0,0.80d0/
      DATA subp/0.482d0,0.480d0,0.428d0,0.473d0,0.456d0,0.405d0,
! subsoil porosity based on texture, from ACRU manual
     c          0.500d0,0.500d0,0.480d0,0.466d0,0.466d0,0.466d0,
     c          0.466d0,0.477d0,0.477d0,0.477d0,0.477d0,0.440d0,
     c          0.440d0,0.440d0,0.440d0/

c --- Get initial input values
C --- INITIALIZE INTERNAL VARIABLES
      BFLOSS = 0.D0

      IF (soilpt.eq.0.d0) THEN !if a porosity value is not
provided, choose subsoil porosity based on texture
          soilpt = subp(isoil)
      else
          soilpt = soilpt
      end if
      storvol=soilpt*zstore*Area*10000.d0 !total volume (m3) of
soil water storage capacity
      IF (JDAY.EQ.1.D0) THEN
          SISTORE(INODE)=(WP+0.5D0*(FC-WP))*storvol
      end if
      SIWater1 = SISTORE(inode) !shallow infiltration water
storage (m3)
      recharge=0.d0
      SIwater2 = 0.d0
      Write(nut,200)SISTORE(inode)
      BFlossfrac = sfrac1(isoil) !baseflow loss fraction
```

```
      Rlossfrac = sfrac2(isoil) !RECHARGE loss fraction
      DpercM3 = Dperc*10*Area !Dperc from ThetaFAO converted to
m3

      SIWC1=SIWater1/storvol !calculate water content (m3/m3) of
soil
      FCm3=FC*storvol !water content (m3) of soil at field
capacity
      maxstore = FCm3
      minstore = WP*storvol

      dstorvol(inode)=storvol
      dmaxstor(inode)=maxstore
      dminstor(inode)=minstore
      drloss(inode)=Rlossfrac
      dSIWater1(inode)=SIWater1

      !   if (jday.eq.1) then
      !      print*,'dpseep',
inode,storvol,maxstore,Rlossfrac,SIWater1
      !      print*,'matrix', dstorvol(inode),dmaxstor(inode),
      ! &     drloss(inode),dSIWater1(inode)
      !      end if

c --- Calculate losses to baseflow and recharge
      IF (SISTORE(INODE).LT.MINSTORE) then !if soil water
content is less than or equal to wilting point, then no water is
lost to baseflow or recharge
          BFloss = 0.d0
          Recharge = 0.d0
      else if (SISTORE(INODE).GT.MAXSTORE) then
          BFloss = (SISTORE(INODE)-MAXSTORE)+BFLOSSFRAC*MAXSTORE
          SISTORE(INODE) = SISTORE(INODE)-BFloss ! Water
remaining after baseflow losses
          Recharge = SISTORE(INODE)*Rlossfrac ! Water lost to
recharge
          DSWC2=SISTORE(INODE)-MINSTORE
          IF (RECHARGE.GT.DSWC2) THEN
              RECHARGE=DSWC2
              SISTORE(INODE)=MINSTORE
          ELSE
              RECHARGE=RLOSSFRAC*SISTORE(INODE)
              SISTORE(INODE)=SISTORE(INODE)-RECHARGE
          END IF
      else
       BFloss = BFLOSSFRAC*SISTORE(INODE)
```

```
       SISTORE(INODE) = SISTORE(INODE)-BFloss ! Water remaining
after baseflow losses
        IF (SISTORE(INODE).GT.MINSTORE) THEN
           Recharge = SISTORE(INODE)*Rlossfrac ! Water lost to
recharge
           DSWC2=SISTORE(INODE)-MINSTORE
              IF (RECHARGE.GT.DSWC2) THEN
                  RECHARGE=DSWC2
                  SISTORE(INODE)=MINSTORE
              ELSE
                  RECHARGE=RLOSSFRAC*SISTORE(INODE)
                  SISTORE(INODE)=SISTORE(INODE)-RECHARGE
              END IF
        END IF
      END IF

c --- Recalculate shallow infiltration water volume by adding
new percolation from ThetaFAO
       SISTORE(inode)= DPercM3+SISTORE(inode)
      IF (SISTORE(INODE).GT.MAXSTORE) THEN
          BFLOSS2 = (SISTORE(INODE)-MAXSTORE)
          SISTORE(INODE)=MAXSTORE
      END IF
C --- Recalculate soil water then calculate recharge accordingly
        SWC2=SISTORE(INODE)/storvol
      !  IF (SWC2.le.wp) then
      !      recharge=0.d0
      !  else
      !    RECHARGE=Rlossfrac*SISTORE(INODE)
      !    SISTORE(INODE)=SISTORE(INODE)-RECHARGE
      !        IF (SISTORE(inode).gt.maxstore) then
      !        RECHARGE = RECHARGE +(SISTORE(INODE)-MAXSTORE) !
any excess water goes to baseflow
      !        SISTORE(INODE) = maxstore
      !        END IF
      !  END IF
      !print*,'rechargedpseep',jday,inode,recharge
      DSIwater = SISTORE(INODE)-SIwater1
      SWC2=SISTORE(INODE)/storvol

      DRECH(JDAY,INODE)=RECHARGE
      DSM2(JDAY,INODE)=DSIwater
      DBF(JDAY,INODE)=BFLOSS
      DTHETA2(JDAY,INODE)=SWC2

C ---------------------------------------
```

```
C Add baseflow water to each timestep over next 24 hours !!
commented out by LW 7.13.2023
C ---------------------------------------
!      BFlossm3 = BFloss/288.d0 !Calculates baseflow lost every
5 minutes (m3)
!      BFlosscms = BFlossm3/300.d0
!      BFlosscfs = BFlosscms*(3.28084d0**3)
!      CALL MREAD(NA,A)
!      DO 20 I=1,288
!          A(I)=A(I)+BFlosscfs
!20    CONTINUE
!      CALL MWRITE(NA,A)



c -- Write outputs --
      write(nut,210)Dpercm3
      write(nut,230)soilpt
      write(nut,240)Zstore
      !write(nut,250)BFloss
      !write(nut,260)BFlosscfs
      !write(nut,270)Recharge
      write(nut,300)SISTORE(inode)

200   format('Initial soil water as Shallow inf
(m3)',10x,'=',f15.1)
210   format('Volume of new shallow infiltration
(m3)',9x,'=',f15.3)
230   format('Subsoil porosity (m3/m3)',24x,'=',f15.3)
240   format('Subsoil storage depth (m)',23x,'=',f15.3)
250   format('Shallow infiltration lost as baseflow
(m3)',6x,'=',f15.3)
260   format('Shallow infiltration lost as baseflow
(cfs)',5x,'=',f15.3)
270   format('Shallow infiltration lost as recharge
(m3)',6x,'=',f15.3)
300   format('Remaining soil water storage (m3)',15x,'=',f15.3)

      end subroutine dpseep
```

```
       SUBROUTINE FINPUT(LISFIL)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCC
C      Create input and output file names from a command line
input string    C
C     NOTE: Maximum lenght of command line string = 50
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCC

      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE

COMMON/NNINOUT/NIET,NIBF,NITM,NITU,NITL,NIWS,NISR,NICK,NIAB,NIIR
,
     &
NOPR,NORO,NOET,NODN,NOSM,NOQI,NODS,NOAB,NOIR,NOQO,NOMO,NOPE,NOWB
,
     & NOSD,NISN,NOIN
      CHARACTER*50 FILENM1
      CHARACTER*75 LISFIL(31)
      CHARACTER*4,SCOD(31)
      CHARACTER*1,DUMMY1
      character*200 linein
      character*1 slash

DATA(SCOD(I),I=1,31)/'idat','oans','osss','iprn','odss','ieto',
     &
'ibfl','itmp','itma','itmi','iwsp','isor','ickm','iabs','iirr',
     &
'oprn','odro','oeta','odng','osmi','oqif','odst','oabs','oirr',
     & 'oqof','odmo','opef','owbl','osed','isno','oinp' /

c*** Command line option to input filename
c*** Comment out the following depending for which system you
compile

      IDAYOUT=0
CWIN32*** Start of Win32 file i/o     ***
CWIN32      slash='\'
CWIN32      INARGS=NARGS()-1
CWIN32      IF (INARGS.EQ.1) THEN
CWIN32          CALL GETARG(1,FILENM1,IFSTATUS)
CWIN32        ELSEIF (INARGS.EQ.2) THEN
CWIN32          CALL GETARG(1,FILENM1,IFSTATUS)
```

```
CWIN32          IDAYOUT=1
CWIN32*** End of Win32 file i/o       ***
cUNIX *** Start Unix file i/o         ***
CUNIX
      slash='/'
CUNIX
      INARGS=IARGC()
CUNIX
      IF (INARGS.EQ.1) THEN
CUNIX
         CALL GETARG(1,FILENM1)
CUNIX
        ELSEIF (INARGS.EQ.2) THEN
CUNIX
           CALL GETARG(1,FILENM1)
CUNIX
             IDAYOUT=1
cUNIX*** End of UNIX file i/o section ***
       ELSE
          WRITE(*,*)
          WRITE(*,105)
          WRITE(*,110)
          WRITE(*,130)
          WRITE(*,140)
          WRITE(*,150)
          WRITE(*,*)
          STOP
      ENDIF

c-----Write welcome message ------------------------------------
---
      write(*,*)
      WRITE(*,160)
      WRITE(*,*)'   @@@ @  @  @@@@ @  @  @@@  @@'
      WRITE(*,*)'  @    @  @  @    @@ @ @    @  @'
      WRITE(*,*)'  @    @  @  @@@  @ @@ @    @@@@'
      WRITE(*,*)'  @    @  @  @    @  @ @    @  @'
      WRITE(*,*)'   @@@  @@   @@@@ @  @  @@@ @  @  June 2023-
v0.4'
      WRITE(*,160)
      WRITE(*,*)'  LINK AND NODE WATERSHED SIMULATION MODEL'
      WRITE(*,160)
      WRITE(*,*)'     R.Munoz-Carpena & Lory Willard'
      WRITE(*,*)'     UFL - USA'
      WRITE(*,*)'     lory.willard@ufl.edu'
      WRITE(*,160)
c      WRITE(*,*)
```

```
C------ create I/O filenames from input string -----------------
--------
c------    or read filenames from a project file ---------------
--------

      ilstr=index(filenm1,'.')
      if (ilstr.gt.0) then
c       *** using project file (.prj or .lis) to read filenames
c       *** mods made 10/27/99, jep - push version to 1.0
c       ***  check to see if extension is .prj or .lis
          ilstr1=index(filenm1,'.prj')
          ilstr2=index(filenm1,'.lis')
          if ((ilstr1.gt.0).or.(ilstr2.gt.0)) then
c          *** fill filename array with safe names
            do 11 i=1,31
              dummy1=scod(i)
                  IF(DUMMY1.EQ.'i') THEN
                     WRITE(LISFIL(I),'(31A)')
     1                 'inputs',slash,'dummy','.',SCOD(I)
               ELSE
                 WRITE(LISFIL(I),'(31A)')
     1                 'output',slash,'dummy','.',SCOD(I)
              ENDIF
   11       continue
c
            open(unit=99,file=filenm1,status='old')
   12       read(99,'(a)',end=18) linein
            lpos=index(linein,'=')
            lstr=len(linein)
            if ((lpos.gt.0).and.(lstr.gt.0)) then
               do 14 jj=1,31
                  lpp = index(linein(1:lpos-1),scod(jj))
                  if (lpp.gt.0) lisfil(jj)=linein(lpos+1:)
   14          continue
            endif
            go to 12
c          ****** done
   18      continue
          else
            WRITE(*,*)
            WRITE(*,105)
            WRITE(*,110)
            WRITE(*,130)
            WRITE(*,140)
            WRITE(*,150)
            WRITE(*,*)
            STOP
```

```
            endif
        else
c       **** rafa's i/o scheme
          ILSTR=INDEX(FILENM1,' ')-1
          DO 101 I=1,31
              DUMMY1=SCOD(I)
              IF(DUMMY1.EQ.'i') THEN
                    WRITE(LISFIL(I),'(31A)')
     1           'inputs',slash,FILENM1(:ILSTR),'.',SCOD(I)
                 ELSE
                    WRITE(LISFIL(I),'(31A)')
     1           'output',slash,FILENM1(:ILSTR),'.',SCOD(I)
              ENDIF
101       CONTINUE
      endif

       write(*,*)'*** Opening '
      DO 102 I=1,31
c            write(*,*)'*** Opening ',lisfil(i)
            write(*,'(70A)')lisfil(i)
102      CONTINUE
      WRITE(*,*)

C---------Open I/O files --------------------------------
c------------Inputs ----------
      OPEN(NDAT,FILE=LISFIL(1),STATUS='OLD')
      OPEN(NIPR,FILE=LISFIL(4),STATUS='OLD')
      OPEN(NIET,FILE=LISFIL(6),STATUS='OLD')
      OPEN(NIBF,FILE=LISFIL(7),STATUS='OLD')
      OPEN(NITM,FILE=LISFIL(8),STATUS='OLD')
      OPEN(NITU,FILE=LISFIL(9),STATUS='OLD')
      OPEN(NITL,FILE=LISFIL(10),STATUS='OLD')
      OPEN(NIWS,FILE=LISFIL(11),STATUS='OLD')
      OPEN(NISR,FILE=LISFIL(12),STATUS='OLD')
      OPEN(NICK,FILE=LISFIL(13),STATUS='OLD')
      OPEN(NIAB,FILE=LISFIL(14),STATUS='OLD')
      OPEN(NIIR,FILE=LISFIL(15),STATUS='OLD')
      OPEN(NISN,FILE=LISFIL(30),STATUS='OLD')
c------------Outputs ----------
      IF(IDAYOUT.EQ.0)THEN
          OPEN(NUT,FILE='NUL',STATUS='unknown')
          OPEN(NSSS,FILE='NUL',STATUS='unknown')
         ELSE
          OPEN(NUT,FILE=LISFIL(2),STATUS='unknown')
          OPEN(NSSS,FILE=LISFIL(3),STATUS='unknown')
       ENDIF
      WRITE(NUT,220)LISFIL(2)
```

```
      WRITE(NSSS,220)LISFIL(3)
      OPEN(NDSS,FILE=LISFIL(5),STATUS='unknown')
      WRITE(NDSS,220)LISFIL(5)
      WRITE(NDSS,901)
      WRITE(NDSS,701)
      OPEN(NOPR,FILE=LISFIL(16),STATUS='unknown')
      WRITE(NOPR,220)LISFIL(16)
      WRITE(NOPR,902)
      WRITE(NOPR,701)
      OPEN(NORO,FILE=LISFIL(17),STATUS='unknown')
      WRITE(NORO,220)LISFIL(17)
      WRITE(NORO,903)
      WRITE(NORO,701)
      OPEN(NOET,FILE=LISFIL(18),STATUS='unknown')
      WRITE(NOET,220)LISFIL(18)
      WRITE(NOET,904)
      WRITE(NOET,701)
      OPEN(NODN,FILE=LISFIL(19),STATUS='unknown')
      WRITE(NODN,220)LISFIL(19)
      WRITE(NODN,905)
      WRITE(NODN,701)
      OPEN(NOSM,FILE=LISFIL(20),STATUS='unknown')
      WRITE(NOSM,220)LISFIL(20)
      WRITE(NOSM,906)
      WRITE(NOSM,701)
      OPEN(NOQI,FILE=LISFIL(21),STATUS='unknown')
      WRITE(NOQI,220)LISFIL(21)
      WRITE(NOQI,907)
      WRITE(NOQI,701)
      OPEN(NODS,FILE=LISFIL(22),STATUS='unknown')
      WRITE(NODS,220)LISFIL(22)
      WRITE(NODS,908)
      WRITE(NODS,701)
      OPEN(NOAB,FILE=LISFIL(23),STATUS='unknown')
      WRITE(NOAB,220)LISFIL(23)
      WRITE(NOAB,909)
      WRITE(NOAB,701)
      OPEN(NOIR,FILE=LISFIL(24),STATUS='unknown')
      WRITE(NOIR,220)LISFIL(24)
      WRITE(NOIR,910)
      WRITE(NOIR,701)
      OPEN(NOQO,FILE=LISFIL(25),STATUS='unknown')
      WRITE(NOQO,220)LISFIL(25)
      WRITE(NOQO,911)
      WRITE(NOQO,701)
      OPEN(NOMO,FILE=LISFIL(26),STATUS='unknown')
      WRITE(NOMO,220)LISFIL(26)
```

```
      WRITE(NOMO,912)
      WRITE(NOMO,701)
      OPEN(NOPE,FILE=LISFIL(27),STATUS='unknown')
      WRITE(NOPE,220)LISFIL(27)
      WRITE(NOPE,913)
      WRITE(NOPE,701)
      OPEN(NOWB,FILE=LISFIL(28),STATUS='unknown')
      WRITE(NOWB,220)LISFIL(28)
      WRITE(NOWB,914)
      WRITE(NOWB,701)
      OPEN(NOSD,FILE=LISFIL(29),STATUS='unknown')
      WRITE(NOSD,220)LISFIL(29)
      WRITE(NOSD,915)
      WRITE(NOSD,701)
      OPEN(NOIN,FILE=LISFIL(31),STATUS='unknown')
      WRITE(NOIN,220)LISFIL(31)
      WRITE(NOIN,916)
      WRITE(NOIN,701)




                            !WRITE(NSSS,701)   !writes file
formatting (======) at top

!WRITE(NSSS,903)(INT(DPRECIP(1,K)),K=1,NRNODES) !writes the node
headers

      !OPEN(NDAT,FILE=LISFIL(1),STATUS='OLD')
      !OPEN(NUT,FILE=LISFIL(2),STATUS='unknown')
      !OPEN(NSSS,FILE=LISFIL(3),STATUS='unknown')
      !OPEN(NIPR,FILE=LISFIL(4),STATUS='OLD')
      !OPEN(NDSS,FILE=LISFIL(5),STATUS='unknown')

105   FORMAT('Name:    cuenca')
110   FORMAT(9x,'(Link and node watershed simulation model)')
130   FORMAT('Usage:   cuenca filename (max 8 characters or
project
     &      name)')
CWIN32 identifier for the simulation
CWIN32
140   FORMAT('Version: 0.3 for Windows -April 2020')
cUNIX identifier for the simulation
cUNIX 140      FORMAT('Version: 3.0.3 for Unix -March 2020')
150   FORMAT('Authors: R.Munoz-Carpena & Lory Willard (UFL)')
160   FORMAT(72('-'))
220   FORMAT('File: ',A40,9x,'CUENCA v0.2, 10/2022')
```

```
701   FORMAT(1X,200('='))
901   FORMAT(17X,'>>>>> DAILY FLOW VOLUME (M^3) <<<<<')
902   FORMAT(17X,'>>>>> DAILY PRECIPITATION (MM) <<<<<')
903   FORMAT(17X,'>>>>> DAILY DIRECT RUNOFF (MM) <<<<<')
904   FORMAT(17X,'>>>>> DAILY ACTUAL ET (MM) <<<<<')
905   FORMAT(17X,'>>>>> DAILY DRAINAGE (MM) <<<<<')
906   FORMAT(17X,'>>>>> DAILY ENDING SOIL MOISTURE (MM) <<<<<')
907   FORMAT(17X,'>>>>> DAILY STREAM INFLOW (MM) <<<<<')
908   FORMAT(17X,'>>>>> DAILY DETENTION STORAGE (MM) <<<<<')
909   FORMAT(17X,'>>>>> DAILY WATER USE (MM) <<<<<')
910   FORMAT(17X,'>>>>> DAILY IRRIGATION (MM) <<<<<')
911   FORMAT(17X,'>>>>> DAILY STREAM OUTFLOW (MM) <<<<<')
912   FORMAT(17X,'>>>>> DAILY FLOW MOVED (MM) <<<<<')
913   FORMAT(17X,'>>>>> DAILY EFFECTIVE PRECIPITATION (MM)
<<<<<')
914   FORMAT(17X,'>>>>> DAILY WATER BALANCE (M^3)<<<<<')
915   FORMAT(17X,'>>>>> DAILY SEDIMENT CONCENTRATION (g/L)
<<<<<')
916   FORMAT(17X,'>>>>> SUMMARY OF ALL INPUTS <<<<<')

      RETURN
      END
```

```
C PROGRAM 17  -  Based on Hromadka book pag 217
C -------------------------------------------------------------
-------------------------
      SUBROUTINE flowby(DAYQI,DAYQO,DAYMO,DAYDS,
     & SSTORE)
C -------------------------------------------------------------
-------------------------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCC
C THIS SUBROUTINE USES A FIVE-MINUTE UNIT EXPLICIT MODEL TO
SIMULATE A FLOWBY BASIN  C
C VARIABLES:
C
C NA:     Stream "A" number. This stream is the one to be
modeled                    C
C NB:     Stream "B" number [0 for moving the excess flow from
stream A to           C
C         a permanent storage; 1 for moving excess flow from
stream A to stream B] C
C QCAP:   Maximum flow-by Q (m^3/s)
C
C TIME1:  Time for Beginning of results (hrs)
C
C TIME2:  Time for End of results (hrs)
C
C INTERNAL VARIABLES
C A(5555): STORAGE MATRIX FOR FLOWS IN MAIN STREAM
C B(5555): STORAGE MATRIX FOR FLOWS IN SECONDARY STREAM
C sstore(100): STORAGE MATRIX FOR FLOWS IN DEAD STORAGE POND
(NB=0)
C NUMBER: NUMBER OF TIME STEPS FOR CALCULATIONS TO RUN
C STORE: DEAD STORAGE
C Z: FLOW AT A GIVEN TIME STEP EQUAL TO FLOW IN MAIN CHANNEL
C X: EXCESS FLOW FROM MAIN CHANNEL THAT MUST BE MOVED TO STORAGE
AT A GIVEN TIME STEP

C GLOBAL STORAGE MATRICES
C DAYQI(I,K): Stores daily streamflow values at each node(mm)
C DAYMO(I,K): Stores daily flow values permanently removed at
each node (mm)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCC
C -------------------------------------------------------------
-------------------------
C   DECLARE VARIABLES
C -------------------------------------------------------------
-------------------------
```

```
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/INPUTS/DATAINP(100,100)
      DIMENSION
A(5555),B(5555),sumqi24(5555),sumA24(5555),sumB24(5555),
     & SUMST24(5555),DAYDS(5555,100),sstore(100)
      DIMENSION DAYQI(5555,100),DAYQO(5555,100),DAYMO(5555,100)
!     EXPORT Hydrograph, Date (hours) StreamA(m^3/s)
StreamB(m^3/s)
C ----------------------------------------------------------------
-------------------------
C READ IN STREAM DATA
C ----------------------------------------------------------------
-------------------------
      NA=INT(DATAINP(JCOUNT,4))
      NB=INT(DATAINP(JCOUNT,5))
      QCAP=DATAINP(JCOUNT,6)
      TIME1=DATAINP(JCOUNT,7)
      TIME2=DATAINP(JCOUNT,8)
      Area=DATAINP(JCOUNT,9)
C --- INITIALIZE VARIABLES ----
      SUMQI24=0.D0
      SUMA24=0.D0
      SUMB24=0.D0
      SUMST24=0.D0
C ----------------------------------------------------------------
-------------------------
C CONVERSION
C ----------------------------------------------------------------
-------------------------
      CALL MREAD(NA,A)
C --- Calculate and write inflow volume (in mm) to storage
matrix
      SUMQI24(1)=0.5D0*A(1)*5.D0*60.D0*0.0283168D0
      DO 10 I=2,288
          SUMQI24(I)=SUMQI24(I-1)+0.5d0*(A(I-1)+
     &                A(I))*5.d0*60.d0*0.0283168d0 !Calculating
the sum by taking average of timesteps, multiplying over time
step to get volume, and converting to cubic meters
10    CONTINUE
      DO 20 J=1,100
            IF (INT(DPRECIP(1,J)).EQ.NZ2) THEN
               INODE=J
```

```
            END IF
20    CONTINUE
      DAYQI(JDAY,INODE)=SUMQI24(288)  ! normalize 24 hr flow to
mm by dividing by area and converting units
      IF(NB.GT.0) CALL MREAD(NB,B)
C --------------------------------------------------------------
-------------------------
C INITIALIZE VARIABLES
C --------------------------------------------------------------
-------------------------
      TIME=0.d0
      NUMBER=INT(A(5555))
      !print*,'numberflowby',number
C      IF (NUMBER.GT.0.D0) THEN
      IF(NUMBER.GT.5555) NUMBER=5555-1 !ADDED 11.14.2022
      STORE=SSTORE(inode)
      STORE1=SSTORE(INODE)
C --------------------------------------------------------------
-------------------------
C READ MODEL DATA
C --------------------------------------------------------------
-------------------------
       WRITE(NUT,901)NA
       WRITE (NUT,905) NA, QCAP
       IF(NB.EQ.0)WRITE(NUT,902)
       IF(NB.NE.0)WRITE(NUT,903)NB
       IF(NB.EQ.0)WRITE(NUT,921)NA,QCAP,NA
       IF(NB.GT.0)WRITE(NUT,923)NB,NA,QCAP,NB,NA
       IF(NB.EQ.0)WRITE(NUT,908)NA,NA
       IF(NB.GT.0)WRITE(NUT,906)NB,NA,NB,NA
C --------------------------------------------------------------
-------------------------
C MODEL FLOWBY EFFECTS
C --------------------------------------------------------------
-------------------------
       IF(NB.GT.0)THEN
199       DO 200 I=1,NUMBER
           Z=A(I)
           ZB=B(I)
           TIME=TIME+.0833333d0
           X=Z-QCAP
           IF(X)198,198,150
150         B(I)=B(I)+X
               A(I)=QCAP
        !Export Hydrograph to a permanent storage
!!198          IF(TIME.LT.TIME1.OR.TIME.GT.TIME2) GO TO 200
!!            WRITE(NUT,909)TIME,ZB,Z,B(I),A(I)
```

```
198        IF(TIME.GE.TIME1.AND.TIME.LE.TIME2)
WRITE(NUT,909)TIME,ZB,
     &         Z,B(I),A(I)
200       CONTINUE
        ELSE
C --------------------------------------------------------------
-------------------------
C MODEL DEAD STORAGE - THIS STORAGE IS PERMANENT AND RESETS TO 0
EACH DAY
C --------------------------------------------------------------
-------------------------
        !STORE=STORE+SSTORE(inode)
         !STORE=STORE+0.D0
          DO 100 I=1,NUMBER
           TIME=TIME+.08333d0
           Z=A(I)
           X=Z-QCAP
c-rmc-IF(x)10,20,30, go to lines 10, 20 or 30 if the value is
<0, 0 or >0, respectively
               IF(X)99,99,50
50             STORE=STORE+X/145.2d0
                  A(I)=QCAP
99
IF(TIME.GE.TIME1.AND.TIME.LE.TIME2)WRITE(NUT,907)TIME,
     &            Z,A(I),STORE
                 IF (NUMBER.LE.288) THEN
                  SUMST24(I)=(STORE*1233.48d0)! volume converted
from ac-ft to m3
                  STORE24=SUMST24(NUMBER)-STORE1*1233.48D0
                  END IF
                  IF (NUMBER.EQ.288) THEN
                      SSTORE(inode)=STORE
                  ENDIF
100       CONTINUE
         END IF
          SUMA24(1)=0.5D0*A(1)*5.D0*60.D0*0.0283168D0
          DO 120 I=2,288
              SUMA24(I)=SUMA24(I-1)+0.5d0*(A(I-1)+
     &            A(I))*5.d0*60.d0*0.0283168d0
              IF (NB.GT.0) THEN
              SUMB24(I)=SUMB24(I-1)+0.5d0*(B(I-1)+
     &                B(I))*5.d0*60.d0*0.0283168d0
              END IF
120       CONTINUE
              NUMB=INT(B(5555))
              IF(NUMBER.GT.NUMB)NUMB=NUMBER
              B(5555)=NUMB
```

```
C --- Assign daily summary outflows/storage to correct arrays
          DAYQO(JDAY,INODE)=SUMA24(288)
          IF (NB.GT.0) THEN
          DAYMO(JDAY,INODE)=SUMB24(288)
          ELSE
          DAYMO(JDAY,INODE)=STORE24
          END IF

C --------------------------------------------------------------
-------------------------
C SAVE RESULTS
C --------------------------------------------------------------
-------------------------
       CALL MWRITE(NA,A)
       IF(NB.GT.0) CALL MWRITE(NB,B)
      !ELSE
      !WRITE(nut,999)
      !ENDIF
C --------------------------------------------------------------
-------------------------
C   OUTPUT - FORMAT
C --------------------------------------------------------------
-------------------------
921   FORMAT(32X, 'INFLOW',/,31X,'(STREAM',I2,') ',/,
     C 3(35X,'|',/),21X,' ---------    |'/,
     C 21X,'|',9X,'|',3X,'|',/,21X,'|',9X,'|<--* < =flowby
Structure',/,
     C 21X, '|  basin  |   | (Maximum flowby Q = ',F5.1,'
CFS)',/,
     C 21X,'| storage |   |',/,21X,' ---------    |',/,
     C  2(35X,'|',/),35X,'V',/,30X,'
STREAM',I2,/,32X,'FLOWBY',/)
923   FORMAT(20X,'     INFLOW              INFLOW',/,
     C 20X,'     (STREAM',I2,')         (STREAM',I2,')',/,
     C 27X,'|',19X,'|',/,27X,'|',19X,'|',/,
     C 27X,'|',4X,'flow excess',4X,'|',/,
     C 27X,'<-----------------------------------* <=flowby
structure',/,
     C 27X,'|',19X,'|',1X,'(flowby Q = ',F8.1,' CFS)',/,
     C 27X,'|',19X,'|',/,27X,'|',19X,'|',/,27X,'|',19X,'|',/,
     C 27X,'V',19X,'V',/,25X,'STREAM',I2,12X,'STREAM',I2,/,
     C 20X,'+ FLOW EXCESS',13X,'FLOWBY',/)
906   FORMAT(/,14X,'FLOWBY BASIN MODELING RESULTS:',//,
     C 11X,' MODEL        INFLOW    INFLOW   OUTFLOW
FLOWBY',/,
     C 11X,' TIME   ',4('   (STREAM',I2,')'),/,
     C 11X,' (HRS)  ',4X,4('    (CFS)  '))
```

```
907   FORMAT(10X,F7.3,F9.1,F10.1,F13.3)
908   FORMAT(/,14X,'FLOWBY BASIN MODELING RESULTS:',//,
     C 11X,' MODEL   STREAM',I2,'   STREAM',I2,'    BASIN',/,
     C 11X,' TIME    INFLOW     FLOWBY      VOLUME',/,
     C 11X,' (HRS)    (CFS)      (CFS)       (AF)')
909   FORMAT(10X,F7.3,3X,4F10.1)
901   FORMAT(//,10X,'MODEL STREAM NUMBER',I2,' FLOWING PAST A',
     C' FLOWBY STRUCTURE:')
902   FORMAT(10X,'FLOW EXCESS IS ASSUMED TO BE PERMANENTLY
STORED.',//)
903   FORMAT(10X,'FLOW EXCESS IS ASSUMED TO BE ADDED TO STREAM
NUMBER',
     C I2,//)
905   FORMAT(10X,'FLOWRATES IN STREAM #',I2,' WHICH ARE GREATER
THAN',/
     C ,10X,F8.1,' CFS ARE ASSUMED TO BE EXCESS FLOWS ')
999   FORMAT(10X,'NO FLOW OCCURRED ON THIS DAY')

      RETURN
      END SUBROUTINE FLOWBY
```

```
C PROGRAM 16  -  Based on Hromadka book pag 210
C --------------------------------------------------------------
------------------------
      SUBROUTINE fthru(DAYDS,DAYQI,DAYQO,DSTORE,VSTORE)
C --------------------------------------------------------------
------------------------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCC
C THIS SUBROUTINE ROUTES FLOW THROUGH A FLOW THRU BASIN
C
C USING FIVE-MINUTE INTERVALS.  EXPLICIT ALGORITHM IS USED.
C
C VARIABLES:
C
C ***** Line 1: NA,DEADS,S0,V0,NBASIN,TIME1,TIME2 *****
C
C NA:     Stream "A" number. This stream is the one to be
modeled                  C
C DEADS:  Dead storage volume (m^3)
C
C S0:     Initial dead storage volume (m^3)
C
C V0:     Initial basin effective volume (above PL of outlet)
(m^3)               C
C NBASIN:    Number of basin data points (zero at D=0).
C
C            Allowable values [4 - 20]
C
C TIME1:       Time for Beginning of results (hrs)
C
C TIME2:      Time for End of results (hrs)
C
C ***** Line 2: BD(I),BQ(I),BV(I),I=1,NBASIN *****
C
C BD(I):       Basin Depth (m). Allowable values [0-76]
C
C BQ(I):       Basin outflow (m^3/s). Allowable values [0-2831]
C
C BV(I):       Basin Volume (m^2-m=m^3). Allowable values [0-
123348184 m^3]       C
C I=1
C
C
C NBASIN: Number of basin data points (zero at D=0).
C
C            Allowable values [4 - 20]
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCC
C          INTERNAL VARIABLES
C QBASIN: FLOW WITHIN BASIN
C VBASIN: VOLUME IN BASIN
C STORE: VOLUME IN DEAD STORAGE OF BASIN
C NUMBER: NUMBER OF TIME STEPS IN PROCESS
C DEADS: DEAD STORAGE VOLUME
C A(K): INFLOW
C DEPTH2: EFFECTIVE DEPTH AT EACH TIME STEP
C OAVG: OUTFLOW AT EACH TIME STEP
C S2: EFFECTIVE VOLUME AT EACH TIME STEP
C DSTORE: STORES FINAL DEAD STORAGE VOLUME FROM PREVIOUS DAY
C VSTORE: STORES FINAL EFFECTIVE VOLUME FROM PREVIOUS DAY
C Area:contributing watershed area, ha
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCC
C --------------------------------------------------------------
------------------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
------------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
!(8.29.18)
     & SNODE(5555,100),STAIL(5555,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/INPUTS/DATAINP(100,100)
      DIMENSION A(5555),dstore(100),vstore(100),
     & sumqi24(5555),sumA24(5555)
      DIMENSION DAYDS(5555,100),DAYQI(5555,100),DAYQO(5555,100)
!DAILY STORAGE
      DIMENSION BD(20),BQ(20),BV(20),AA(20),BB(20)
C --------------------------------------------------------------
------------------------
      V0=0.d0
      S0=0.d0
      SUMA24=0.D0
      SUMQI24=0.D0
      DO 5 L=1,100
          IF (INT(DPRECIP(1,L)).EQ.NZ2) THEN
            INODE=L
          END IF
5     CONTINUE
      NA=INT(DATAINP(JCOUNT,4))
```

```
      DEADS=DATAINP(JCOUNT,5)
      IF (JDAY.EQ.1) THEN
          S0=DATAINP(JCOUNT,6)
          V0=DATAINP(JCOUNT,7)
      ELSE
          S0=DSTORE(inode)
          V0=VSTORE(inode)
      END IF
      NBASIN=INT(DATAINP(JCOUNT,8)) ! This value varies from 4-
20 and is multiplied by three
      TIME1=DATAINP(JCOUNT,9)
      TIME2=DATAINP(JCOUNT,10)
      AREA=DATAINP(JCOUNT,11)
      JMAX=10+3*NBASIN
      I=0
      J=10
      DO WHILE (I.LT.NBASIN)
          I = I+1
          J = J+1
          BD(I)=DATAINP(JCOUNT,J)
          !PRINT *, 'test BD',BD(I)
          J=J+1
          BQ(I)=DATAINP(JCOUNT,J)
          !PRINT *,'BQ',BQ(I)
          J=J+1
          BV(I)=DATAINP(JCOUNT,J)
          !PRINT *,'BV',BV(I)
      END DO


C      I1=DATAINP(JCOUNT,14)
C      NBASIN2=DATAINP(JCOUNT,15)
      WRITE(NUT,901)NA,DEADS,S0,V0
      WRITE(NUT,903)
      WRITE(NUT,905)(I,BD(I),BQ(I),BV(I),I=1,NBASIN)
      WRITE(NUT,921)NA,NA
C -------------------------------------------------------------
-----------------------
C   CONVERSION
C -------------------------------------------------------------
-----------------------
C!      J1=1
C!      DO 201 I=1,NBASIN ! (marco)
C!      BD(I)=BD(I)/(0.3048d0) !To obtain feet
C!      BQ(I)=BQ(I)/(0.3048d0**3) !To obtain cubic feet per
second
```

```
C!      BV(I)=BV(I)/(1233.8184) !To obtain acre-feet from m^3
(Basin Storage)
C!      J1=J1+3
C!201   CONTINUE ! IF I active the DO
C --------------------------------------------------------------
------------------------
C READ IN STREAM NUMBER NA
C --------------------------------------------------------------
------------------------
      CALL MREAD(NA,A)
C --- Calculate and write inflow volume (in mm) to storage
matrix
      SUMQI24(1)=0.5D0*A(1)*5.D0*60.D0*0.0283168D0
      DO 7 I=2,288
          SUMQI24(I)=SUMQI24(I-1)+0.5d0*(A(I-1)+
     &        A(I))*5.d0*60.d0*0.0283168d0 !Calculating the sum
by taking average of timesteps, multiplying over time step to
get volume, and converting to cubic meters
             IF(JDAY.EQ.5.AND.NZ2.EQ.303) THEN
              print*,i,A(i),SUMqi24(i)
               END IF
7      CONTINUE
      DAYQI(JDAY,INODE)=SUMQI24(288) ! Daily inflow volume is
equal to the sum at timestep 288
C --------------------------------------------------------------
------------------------
C INITIALIZE VARIABLES
C --------------------------------------------------------------
------------------------
      I=1
      QBASIN=BQ(NBASIN)
      NB=NBASIN-1
      VBASIN=BV(NBASIN)
      TIME=0.d0
      STORE=S0
      VOLUME=V0
      NUMBER=INT(A(5555))
!      IF ((NUMBER.GT.0.D0).OR.(V0.GT.0.D0)) THEN
          IF (NUMBER.EQ.0.D0) THEN
              NUMBER=300
          END IF
       ZERO=0.d0
       WRITE(NUT,908)
C --------------------------------------------------------------
------------------------
C   MODEL DEAD STORAGE
```

```
C ----------------------------------------------------------------
------------------------
       IF((DEADS.NE.0.d0).OR.(S0.LT.DEADS)) THEN
           STORE=S0
           DO 220 I=1,NUMBER
               STORE=STORE+A(I)/145.2d0
               TIME=TIME+.083333d0
               X=STORE-DEADS
               IF(X)10,10,155
10          IF(TIME.GE.TIME1.AND.TIME.LE.TIME2)
WRITE(NUT,907)TIME,
     &        STORE,A(I),ZERO,ZERO,ZERO
                 A(I)=0.d0
               IF (NUMBER.LE.288) THEN
                  IF(I.EQ.288) THEN
                    DSTORE(inode)=STORE
                  END IF
                  IF(I.EQ.NUMBER) THEN
                    DAYDS(JDAY,INODE)=(STORE-S0)*1233.48184d0 !
volume converted from ac-ft to m3
                  !PRINT*,'HERE1',JDAY,NZ2
                  END IF
              END IF
220        CONTINUE
C ALL FLOW HELD IN BASIN
           GO TO 2000
C DEAD STORAGE REMAINING
155        ATEMP=A(I)
           A(I)=X*145.2d0
           ATEMP=ATEMP-A(I)
           STORE=DEADS
           TIME=TIME-.08333d0
           IF(TIME.GE.TIME1.AND.TIME.LE.TIME2) THEN
               WRITE(NUT,930)ATEMP,A(I)
               DAYDS(JDAY,INODE)=(ATEMP)*1233.48184d0
!DAYDS(JDAY,INODE)=(ATEMP)*1233.48d0
              !PRINT*,'HERE2',JDAY,NZ2
          END IF
C ROUTE THRU BASIN
C FLOW THRU BASIN MODE.-
        ELSE
100       VOLUME=V0
       END IF
C FIND INITIAL BASIN DEPTH AND OUTFLOW
       DO 115 II=1,NB
           IF(VOLUME.LT.BV(II+1))GO TO 116
115   CONTINUE
```

```
114      TI=TIME+.083333d0
C -------------------------------------------------------------
-----------------------
       WRITE(NUT,909)TI
       II=NB
116      TEMP=(VOLUME-BV(II))/(BV(II+1)-BV(II))
       D0=BD(II)+TEMP*(BD(II+1)-BD(II))
C -------------------------------------------------------------
-----------------------
C GET INITIAL VALUES
C -------------------------------------------------------------
-----------------------
       O2=0.d0
       S2=0.d0
       S1=BV(II)+TEMP*(BV(II+1)-BV(II))
       O1=BQ(II)+TEMP*(BQ(II+1)-BQ(II))
       CON=60.d0/43560.d0*5.d0/2.d0
       DO 1011 K=1,NBASIN
       AA(K)=BV(K)-BQ(K)*CON
       BB(K)=BV(K)+BQ(K)*CON
!!1011  BB(K)=BV(K)+BQ(K)*CON
1011      CONTINUE
       AA(1)=BB(1)
       CON=CON*2.d0
       ATEMP=S1-O1*CON/2.d0
       DO 1000 K=I,576
           QQ=CON*A(K)
           TEMP=QQ+ATEMP
           CALL SEE(TEMP,B1,B2,I1,I2,NUT,BB,NBASIN,TIME) !
Original from the book
           RATIO=(TEMP-B1)/(B2-B1)
           DEPTH2=BD(I1)+RATIO*(BD(I2)-BD(I1))
           S2=BV(I1)+RATIO*(BV(I2)-BV(I1))
           O2=BQ(I1)+RATIO*(BQ(I2)-BQ(I1))
           ATEMP=S2-O2*CON/2.d0
           TIME=TIME+.0833333d0
           OAVG=(O1+O2)/2.d0
           O1=O2
           IF (K.EQ.288) THEN
              DSTORE(inode)=STORE
              VSTORE(inode)=S2
              !PRINT*,'DEADS,S2,VO,SO',DEADS,S2,V0,S0
              DAYDS(JDAY,INODE)=(DEADS+S2-V0-S0)*1233.48184d0 !
volume converted from ac-ft to m3
              !PRINT*,'HERE3',JDAY,NZ2
           ENDIF
```

```
           !IF (K.GT.1) THEN   !!CHANGED FROM IF (I.GT.1) BY
LORY ON 1/31/2022
           !  SUMA24(K) = SUMA24(K-
1)+0.5d0*(OAVG)*5.d0*60.d0*0.0283168d0
           !END IF

IF(TIME.GE.TIME1.AND.TIME.LE.TIME2)WRITE(NUT,907)TIME,DEADS,
     &        A(K),DEPTH2,OAVG,S2
              A(K)=OAVG

1000    CONTINUE
        !IF (NUMBER.GE.288) THEN
        !       DAYQO(JDAY,INODE)=SUMA24(288)
        !ELSE
        !       DAYQO(JDAY,INODE)=SUMA24(NUMBER)
        !END IF
           !A(5555)=576    !! REMOVED by lw 11.14.2022 while
testing how number is affecting flow of tails
2000      CONTINUE
       CALL MWRITE(NA,A)

C --- CALCULATE OUTFLOW WATER BALANCE TERM
        SUMA24(1)=0.5D0*A(1)*5.D0*60.D0*0.0283168D0

      DO 3000 K=2,288
              SUMA24(K) = SUMA24(K-1)+0.5d0*(A(K)+A(K-1))
     &         *5.d0*60.d0*0.0283168d0
          IF(JDAY.EQ.5.AND.NZ2.EQ.303) THEN
              print*,K,A(K),SUMA24(K)
          END IF
3000  CONTINUE

      DAYQO(JDAY,INODE)=SUMA24(288)
C --------------------------------------------------------------
------------------------
C   OUTPUT - FORMAT
C --------------------------------------------------------------
------------------------
901   FORMAT(/,10X,'ROUTE RUNOFF HYDROGRAPH FROM STREAM NUMBER:
',I2,
     C /,10X,'THROUGH A FLOW-THROUGH DETENTION BASIN',/,
     C  10X, 'USING FIVE-MINUTE UNIT INTERVALS:',/,
     C  10X, 'SPECIFIED BASIN CONDITIONS ARE AS FOLLOWS:',/,
     C  10X, 'DEAD STORAGE (AF) = ',F44.3,/,
     C  10X, 'SPECIFIED DEAD STORAGE (AF) FILLED = ',F27.3,/,
     C  10X, 'SPECIFIED EFFECTIVE VOLUME (AF) FILLED ABOVE
OUTLET = ',
```

```
     C  F10.3)

903   FORMAT(//,10X,' BASIN DEPTH VERSUS OUTFLOW AND STORAGE ',
     C  'INFORMATION:'
     C  ,//,11X,' INTERVAL    DEPTH   OUTFLOW    STORAGE'
     C  ,/,11X,'  NUMBER      (FT)      (CFS)      (AF)')

905   FORMAT(10X,I7,2X,F10.2,F10.2,F10.3)
907   FORMAT(10X,F7.3,F13.3,F9.1,F10.2,F9.1,F11.3)
921   FORMAT(///,20X,'
INFLOW',/,20X,'(STREAM',I2,')',/,3(25X,'|',/)
     C  25X,'V',15x,'Effective depth',/,
     C  20X,  ' -----------',
     C  9X,' | (and Volume)',/,
     C  20X,'|',11X,'|',4X,'|
|',/,20X,'|',11X,'|',4X,'|....',
     C  'V........',/,
     C  20X,  '| detention |<-->|        outflow',/,
     C  20X,  '|   basin   |    |........._____',/,
     C  20X,  ' -----------     |    ^    |    \',/,
     C  20X,  '      |           | storage |   basin outlet',/,
     C  20X,  '      V           ---------',/,
     C  22X,  'OUTFLOW',/,21X,'(STREAM',I2,')',//)

908   FORMAT(11X,'BASIN ROUTING MODEL RESULTS(5-MINUTE
INTERVALS):'
     C  ,//,11X,'TIME     DEAD-STORAGE INFLOW   EFFECTIVE
OUTFLOW ',
     C        'EFFECTIVE'
     C  ,/,11X,'(HRS)     FILLED(AF)  (CFS)    DEPTH(FT)  (CFS)
',
     C' VOLUME(AF)')

930   FORMAT(/,11X,'DEAD STORAGE FILLED WITH UNIT INFLOW(CFS) =
',
     C  F15.1,/,11X,'REMAINING UNIT FLOW IS = ',F34.1,' CFS ',/)
909   FORMAT(10X,F7.3,5X,
     C '*BASIN CAPACITY EXCEEDED; BASIN DATA IS EXTRAPOLATED*')
999   FORMAT(10X,'NO FLOW OCCURRED ON THIS DAY')
C --------------------------------------------------------------
-----------------------
C END PROCESS
C --------------------------------------------------------------
-----------------------

       RETURN
       END SUBROUTINE fthru
```

```
      SUBROUTINE
GAINPUTS(NA,Area,jstype,D,pL,Y,ITCTYPE,isoil,ek,
     C                  cfact,pfact,dp,ieroty,xIa,om,
     C
uFC,uWP,ZR,PFRAC,HM,dtheta,soilpt,Zstore)

C     SUBROUTINE TO READ INPUTS FOR THE GASH PROGRAM
C-------------------------------------------------------------
--------------
C   VARIABLE DEFINITIONS
C ITCTYPE: Method for calculating time of concentration
C Y: Watershed slope, m/m
C pL: Longest flow path, m
C Area: Area, ha
C jstype: SCS storm type (I, IA, II, III, or 'user')
C P: Precipitation (mm)
C D: Storm duration (hr)
C IxA: initial abstraction (always 0 for green ampt)

      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)
      common /gampar/ vsatk,sav,wcsat,wcini,bm,deltim,stmax

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/INPUTS/DATAINP(100,100)
      dimension dtheta(5555,100)


      DO 20 J=1,100
          IF (INT(DPRECIP(1,J)).EQ.NZ2) THEN
              INODE=J
          END IF
20    CONTINUE
C -------------------------------------------------------------
--------------
C   READ INPUT DATA
C -------------------------------------------------------------
--------------

      NA=INT(DATAINP(JCOUNT,4))
      Area=DATAINP(JCOUNT,5)
      JSTYPE=INT(DATAINP(JCOUNT,6))
      D=DATAINP(JCOUNT,7)
      pL=DATAINP(JCOUNT,8)
      Y=DATAINP(JCOUNT,9)
```

```
      ITCTYPE=INT(DATAINP(JCOUNT,10))
C --------------------------------------------------------------
------------------------
c READ INPUTS: Soil Erosion Calculations:
C --------------------------------------------------------------
------------------------
c   isoil = soil type (integer), see musle.f data for list soil
types
c   ek     = soil erodibility
c   cfact  = C factor
c   pfact  = P factor
c   dp  = sediment size (d50) in cm. If dp= -1 dp is set based
on "isoil
C --------------------------------------------------------------
------------------------
      isoil=INT(DATAINP(JCOUNT,11))
      ek=DATAINP(JCOUNT,12)
      CFACT=DATAINP(JCOUNT,13)
      PFACT=DATAINP(JCOUNT,14)
      DP=DATAINP(JCOUNT,15)
C --------------------------------------------------------------
------------------------
c     Convert dp to um
C --------------------------------------------------------------
------------------------
      dp=dp*10000.d0
C --------------------------------------------------------------
------------------------
c-  ieroty = select method to estimate storm erosion:
c-    0 or not present = Foster's method for R-factor
c-    1 = Using Williams R-factor
c-    2 = Using R-factor from GLEAMS with daily rainfall
C --------------------------------------------------------------
------------------------
      ieroty=INT(DATAINP(JCOUNT,16))
      IF ((ieroty.lt.3).and.(ieroty.ge.0)) GO TO 24
22       ieroty=1
24       CONTINUE
C---------------------------------------------------------------
---
c    Read Green-Ampt specific
C deltim: timestep for analysis, minutes
C vsatk:vertical saturated K (cm/h)
C sav:average suction at wetting front, Sav (cm)
C wcsat:Saturated water content (cm3/cm3)
C wcini:Initial water content at start of storm (cm3/cm3)
C stmax:Max surface storage (cm), typically 0
```

```
C D: Storm duration (hr)
C IxA: initial abstraction (always 0 for green ampt)
C---------------------------------------------------------------
---

      DELTIM=DATAINP(JCOUNT,17)
      VSATK=DATAINP(JCOUNT,18)
      SAV=DATAINP(JCOUNT,19)
      WCSAT=DATAINP(JCOUNT,20)
      IF (JDAY.EQ.1) THEN
          WCINI=DATAINP(JCOUNT,21)
      ELSE
          WCINI = DTHETA(JDAY-1,INODE)
      END IF
      !print*,'wcini',wcini
      STMAX=DATAINP(JCOUNT,22)

      DELTIM=deltim/60.d0
      xIa=0.D0 !always 0 in GA process
      bm=wcsat-wcini ! Initializing bm, don't add to inputs and
possibly clean up common block ! LW 7.1.2022
C --------------------------------------------------------------
------------------------
c om = % soil organic matter, read IF ek <0
C --------------------------------------------------------------
------------------------
      om = 2.0d0
      IF (ek.lt.0.d0) THEN
         !READ(NDAT,*,END=32)om
          om=DATAINP(JCOUNT,23)
         END IF
C---------------------------------------------------------------
-----------------------
C ThetaFAO inputs
C---------------------------------------------------------------
-----------------------
c   uFC(m3/m3): top soil field capacity water content (read
internally or provided by user when isoil=-1)
c   uWP(m3/m3): top soil wilting point water content (read
internally or provided by user when isoil=-1)
c   Zr(m): maximum grass root zone depth (typical values (0.5-
1.5 m)
c   pfrac[-]: fraction of easily estractable water (typical 0.6
for Bermuda grass)
c   Hm(m): height of vegetation (from VFSMOD *.igr file,
H(cm)/100)
```

```
c   soilpt(m3/m3): subsoil porosity (select 0 if you want it to
be based on texture)
c   Zsoil(m): Difference in highest land surface elevation and
streambed elevation at node (m)
c   Zstore (m): Subsoil storage depth (Soil depth - rooting
zone)

      uFC=DATAINP(JCOUNT,24)
      uWP=DATAINP(JCOUNT,25)
      ZR=DATAINP(JCOUNT,26)
      PFRAC=DATAINP(JCOUNT,27)
      HM=DATAINP(JCOUNT,28)
      soilpt = DATAINP(JCOUNT,29)
      Zsoil = DATAINP(JCOUNT,30) !difference in surface and
stream elevation
      Zstore = Zsoil-ZR
C -------------------------------------------------------------
------------------------

C -------------------------------------------------------------
--------------
C   READ NUMBER OF DAYS AND NODES FOR INPUTS
C -------------------------------------------------------------
--------------
!      READ(NIPR,*)NRDAYS,NRNODES
!      READ(NIPR,*) !SKIP LINE
!      READ(NIPR,*)N,(DPRECIP(1,K),K=1,NRNODES),
!     & (DETO(1,K),K=1,NRNODES),(DBF(1,K),K=1,NRNODES),
!     & (DSM(1,K),K=1,NRNODES),(DAB(1,K),K=1,NRNODES),
!     & (DIRR(1,K),K=1,NRNODES)
C -------------------------------------------------------------
---------
C  READ STARTING NODE NUMBER
C -------------------------------------------------------------
---------
!      NZ2=DPRECIP(1,1)
C -------------------------------------------------------------
---------
C  READ RAINFALL, ET, AND IRRIGATION FOR THE DAY
C -------------------------------------------------------------
---------
C      READ(NIPR,*)(DPRECIP(2,K),K=1,NRNODES),
C     & (DETO(2,K),K=1,NRNODES), (DIRR(2,K),K=1,NRNODES)
!      READ(NIPR,*)N,(DPRECIP(2,K),K=1,NRNODES),
!     & (DETO(2,K),K=1,NRNODES),(DBF(2,K),K=1,NRNODES),
!     & (DSM(2,K),K=1,NRNODES),(DAB(2,K),K=1,NRNODES),
!     & (DIRR(2,K),K=1,NRNODES)
```

```
!      K=1 !! need to update with cuenca incorporation
!      Pinit=DPRECIP(2,K)
!      ET=DETO(2,K)
!      BASEF=DBF(2,K)
!      WCINIT=DSM(2,K)
!      ABSTR=DAB(2,K)
!      PIRR=DIRR(2,K)
!
!!c Soils inputs
!!      deltim =5.d0/60.d0
!!      vsatk=.044d0
!!      sav=22.4d0
!!      wcsat=.499d0
!!      wcini=.25d0
!!      stmax=.5d0
!!      !Yolo Clay - Test Case
!      CALL PCALC(Pinit,P,ET,ABSTR,PIRR)
!
      RETURN
      END SUBROUTINE
!
!
!      SUBROUTINE PCALC(Pinit,P,ET,ABSTR,PIRR)
!!C ---------------------------------------------------------------
----------------
!! This subroutine calculates effective rainfall by
incorporating ET, Baseflow, initial moisture content,
!C     surface water abstractions, and irrigation into effective
rainfall
!!C ---------------------------------------------------------------
-----------
!      IMPLICIT DOUBLE PRECISION (a-h, o-z)
!
COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
!
!      P=Pinit-ET-ABSTR+PIRR
!      RETURN
!      END SUBROUTINE
```

```
     subroutine gampt(ndtime,D,Q,CINF,AREA)
c-----------------------------------------------------------------
--
c  Test for grn-ampt subs, jep, ver. 0.8 rmc
|
c  This program was written to illustrate the Green-Ampt Eqn.
|
c        for modeling unsteady rainfall.  I wrote this program
|
c        for my BAE 463/573 Introduction to Surface Water
Quality|
c        Modeling Class.
|
c Reference: Chu, S. T. 1978. Infiltration during unsteady rain.
|
c            Water Resources Research. 14(3):461-466.
|
c
|
c  I retain the ownership rights to this program. However, you
|
c     are welcome to  modify, use, redistribute, etc as long as
|
c     you do not charge for the program. A reasonable charge for
|
c     distribution and handling costs are appropriate.
|
c  If you have any questions, you can contact me via:
|
c      email: john_parsons[at-@]ncsu.edu
|
c
|
c-----------------------------------------------------------------
--
c    3/23/94 original version                          |
c    2/03/00 rmc, modified tp, tpp and                 |
c            set f=R during no-pond conditions         |
c    3/01/00 changed time array to times since         |
c    2/03/00 rmc, ver. 0.2, modified tp, tpp and       |
c            set f=R during no-pond conditions         |
c    3/02/00 rmc, ver. 0.3, added check for R>Ks       |
c            for no-ponding conditions  and docs.      |
c            time is a keyword in f95                  |
c    6/25/03 added project file stuff to enable        |
c            use of program with vb shell, jep         |
c   10/16/03 fixed bug in total infiltration during    |
```

```
c              periods with rainfall rate < f,        |
c              some code cleanup for readability jep  |
c     2/7/04 fixed bug with check of infiltration rate |
c              being smaller than rainfall rate, jep  |
c    6/16/05 bug in 1st time ponding, neg tnp set to  |
c              zero, also fixed format on K output    |
c                                                     |
c                                                     |
c-------------------------------------------------------
c   Variables
c     deltim: timestep (hr)
!     Rfi(i): rainfall intensity at each time step (m/s)
!     Rainint(i): rainfall intensity at each time step (cm/hr)
!     Rti1(i): starting time of each hyetograph time step
!     Rti(i): ending time of each hyetograph time step
!     Ndtime: number of timesteps in hyetograph
!     Nrain: Number of timesteps in infiltration calculations
!     Sttime(i): start time of each time step in GA
!     Endtim(i): end time of each time step in GA
!     Rawrfi(i): rainfall intensity of each time step in GA
(cm/hr)
!     times(ntimes): sttime(1)
!     bf(ntimes): Array that holds cumulative infiltration (cm)
!     f(ntimes): Array that holds infiltration rate at each time
step (cm/hr)
!     stor(ntimes): Surface storage at a each time step (cm)
!     ro(ntimes): Runoff (cm)
!     prec(ntimes): Total rainfall (cm)
!     rint(ntimes):rawrfi(1) and rrfi at each time step
!     ttp(ntimes):Array that holds time to ponding
!     ttpp(ntimes):Array that holds “to” for calculating
drawdown time
!     fpp(ntimes): Array that holds instantaneous infiltration
(cm/h)
!     tp: time to ponding
!     tpp: equivalent to “to” in Mein and Larson, 1973; used in
calculation of time it takes for ponding to end (bfp - sav * bm
* log (1.0+bfp/(sav*bm)))/vsatk)
!     tnp:
!     wbalck: Cumulative infiltration + runoff (cm)
!     ropeak: Peak runoff rate (cm/h)
!     rotpk: Time when peak runoff rate occurs (h)
!     rfpeak: Peak rainfall intensity (cm/h)
!     rftpeak: Time when peak runoff rate occurs
!
!     Variables inside ponding assessment loop:
!     tstart: Start time at calculation step
```

```
!     tend: End time at calculation step
!     dper: Difference between start and end time
!     rrfi: rainfall intensity at calculation step (cm/hr)
!     train: Total rainfall that fell during calculation step
(cm)
!
!     ipond: binary indicator of ponding
!     fp: instantaneous infiltration (cm/h)
!     Vsatk: vertical saturated K (cm/h)
!     Sav: average suction at wetting front, Sav (cm)
!     Wcsat: Saturated water content (cm3/cm3)
!     Wcini: Initial water content at start of storm (cm3/cm3) -
based on AMC?
!     Stmax: Max surface storage (cm)
!     Bm: wcsat -- wcini, i.e. amount of water soil can hold
until it reaches saturation/ponding, "M" in GA equations
!     Bfp: bfp = sav * bm*vsatk/(fp-vsatk), amount of rainfall
needed to induce ponding (cm), infiltration potential
!
!     Subroutine nopond:  Updates arrays for all values at the
timestep, when no ponding is occurring to time tp
!     Stor(ntimes) = 0
!     Ro(ntimes) equal to previous timestep
!
!     Delinf: infiltration occurring over a time step (cm)
!     Dterr: variable used to make sure we are end of timestep
period
!
!     Subroutine pndinf: calculates amount of ponding occurring
from tp to tend
!     Water: Amount infiltrated in timestep + previous amount
stored
!     Bbf: guess for bigf based on previous bigf and amount of
water
!     Ctime: time at timestep to pass to sschu

!     Subroutine SSCHU: Using newton□s method on Chu□s equation
to determine bff (cumulative infiltration)
!     Delinf: Difference between Current and previous timestep
bigF (i.e. infiltration that occurring at this timestep'
C --------------------------------------------------------------
----------------------------------------------------------

      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
     !character*80  lfile, rfile, sfile, ofile, jfiles(3)
     dimension sttime(5000),endtim(5000),rawrfi(5000),rti1(5000)
```

```
     character*80 linein
     !character*5 scod(3)
     !data (scod(i),i=1,3)/'soils','rainf','outpt'/
      common /gampar/ vsatk,sav,wcsat,wcini,bm,deltim,stmax

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE

COMMON/rain/rfix,rti(5000),rfi(5000),rcum(5000,2),ref(5000,2),nc
um
     common /gamp1/ tp,tpp,fp
     common /gamp2/ ttp(5000),ttpp(5000),fpp(5000)
     common /grunoff/ bf(5000),f(5000), stor(5000),
     &    ro(5000),prec(5000),rint(5000)
      common /deltaro/ dro(5000),times(5000)
      dimension rainint(5000)

c -----------------------------------------------------------------
----
c Initialize internal arrays
c -----------------------------------------------------------------
----
      DO 4 I=1,5000
          bf(I)=0.D0
4     CONTINUE

c -----------------------------------------------------------------
----
c Prepare inputs passed from hyetgh.f
c -----------------------------------------------------------------
----

      DO 5 i=1,5000
c --lw 5.30.2022-- convert rfi(i) from m/s to cm/hr
          rainint(i)=rfi(i)*3600.d0*100.d0
C --lw 5.30.2022 -- create rti1(i) array to be start time
          IF (i.gt.1) THEN
             rti1(i)=rti(i-1)
          END IF
  5   CONTINUE

! CODE FOR READING INPUTS FROM FILE
!    inargs = iargc()
!    if (inargs.gt.1) then
!       call getarg(1, sfile)
!       call getarg(2, rfile)
!       print *
!       print *,' Working with Soils data    =', sfile
```

```
!       jfiles(1) = sfile
!       print *,' Working with Rainfall data =',rfile
!       jfiles(2) = rfile
!       jfiles(3) = 'null'
!    elseif (inargs.eq.1) then
!       call getarg(1,lfile)
!c  gpj project file for use with vb
!         open(unit=99,file=lfile, status='old')
!  648       read(99,'(a)', end=650) linein
!         lpos=index(linein,'=')
!       lstr=len(linein)
!       if ((lpos.gt.0).and.(lstr.gt.0)) then
!          do 649 jj=1,3
!             lpp = index(linein(1:lpos-1),scod(jj))
!             if (lpp.gt.0) jfiles(jj)=linein(lpos+1:)
!  649       continue
!         endif
!       go to 648
!c  done
!  650   continue
!        close(99)
!    else
!       call documnt
!    print *, "Enter Soils file:"
!    read (*,'(A)') sfile
!    print *, "Enter Rainfall File:"
!    read (*,'(A)') rfile
!         jfiles(1) = sfile
!       jfiles(2) = rfile
!       jfiles(3) = 'null'
!      print *,"Soils:",sfile," Rain:",rfile
!c      stop
!    endif
!c   open (unit=10, file=rfile, status='old')
!    open (unit=10, file=jfiles(2), status='old')

c --lw 5.30.2022 -- reassign input variables to arrays used in
GAmpt Subroutine
      nrain=ndtime-1
      DO 6 i=1,Nrain
          sttime(i)=rti1(i+1)
          endtim(i)=rti(i+1)
          rawrfi(i)=rainint(i+1)
          IF (endtim(i).gt.D) THEN
              sttime(i)=0.d0
          END IF
c          PRINT*,STTIME(I),ENDTIM(I),RAWRFI(I)
```

```
  6   CONTINUE

     nrain = 0
      nrain = ndtime-1
 !  10  read(10,*,end=20) rawstt,rawend, rawrai
     !      nrain=nrain+1
     !      sttime(nrain)=rawstt
     !      endtim(nrain)=rawend
     !      rawrfi(nrain)=rawrai
     !      go to 10
 !  20  continue
 !
     !close (10)
c
c    open (unit=11, file=sfile, status='old')
     !open (unit=11, file=jfiles(1), status='old')
     ! read(11,*) deltim, timoff
     ! read(11,*) vsatk, sav, wcsat, wcini
        bm = wcsat - wcini
c     read(11,*) stmax

c    close (11)

 !     lpos=index(jfiles(3),'null')
     !if (lpos.eq.0) then
     !  open(unit=22, file=jfiles(3))
     !else
     !  open(unit=22, file='gampout22.txt')
     !endif
     !open(unit=22, file='gampout22.txt')
c****************************************
c*  write header info for program and   *
c*  output the inputs for checking      *
c****************************************
     call outinp
c
c
     ntimes=1
     times(ntimes)=sttime(1)
     bf(ntimes)=0.d0
     f(ntimes)=0.d0
     stor(ntimes)=0.d0
     ro(ntimes)=0.d0
     prec(ntimes)=0.d0
     rint(ntimes)=rawrfi(1)
     ttp(ntimes)=0.d0
     ttpp(ntimes)=0.d0
```

```
     fpp(ntimes)=0.d0
     ipond=0
     fp=0.d0
     tp=0.d0
     tpp=0.d0
     tnp=0.d0
      bfp=0.d0
c
     do 100 jj=1,nrain
       tstart=sttime(jj)
       tend = endtim(jj)
       dper = tend - tstart
       rrfi=rawrfi(jj)
!      train = rrfi * dper

c**********************************
c* Is there ponding in this period?
c* At the start of the period we need to check two
c* conditions:
c*   1)  Not ponded at the start of the period, then
c*       a) continue not ponded
c*       b) become ponded during the period
c*   2)  if the period starts with ponding, then
c*       a) ponded condition for the entire period.
c*       b) ponding ceases during the period
c************************************************************
c
c*************
c* Condition 1
c*************
     if (ipond.le.0) then
c         *********************************
c         *  find time to ponding         *
c         *********************************
       if (fp.gt.0.d0.and.rrfi.ge.fp) then     ! if
instantaneous inf > 0 and rainfall intensity .ge. instantaneous
infiltration
          tp=tstart
          bfp = sav * bm*vsatk/(fp-vsatk)
         elseif (rrfi.gt.vsatk) then         ! if rainfall
intensity > vertical ksat
          bfp = sav * bm /((rrfi/vsatk)-1.d0)
          tp = bfp/rrfi
             tp=tp+tstart
         else
          tp=9999
       endif
```

```
c         *********************************
c         *   find tpp                    *
c         *********************************
       tpp = (bfp - sav * bm * log (1.d0+bfp/(sav*bm)))/vsatk
       if (fp.gt.0.d0.and.rrfi.lt.fp) tpp=tpp+tstart
       if (tp.gt.tend) then
c           ************************************
c           * a. no ponding during this period *
c           ************************************
           tp=9999
           tpp=9999
           call nopond(tstart,tend,rrfi,ntimes)
          ipond=0
       else
c           **********************************
c           * b. ponding at tp               *
c           **********************************
           call nopond(tstart,tp,rrfi,ntimes)
c               ********************************
c            * ponded from tp on to tend    *
c            ********************************
           call pndinf(tp,tend,rrfi,tp,tpp,fp,ntimes)
           ipond=1
           endif
     else
c          ***************
c          * Condition 2 *
c          ***************
c          ***********************************
c          *  find time to infiltrate fnp    *
c          ***********************************
        frate=f(ntimes)
        if (rrfi.lt.frate) then
             amtinf= bf(ntimes)+stor(ntimes)
             call newtnp(tstart,tend,tnp,tp,tpp,rrfi,amtinf)
        else
c            *****************************************
c            *  will not loose ponding, set tnp>tend *
c            *****************************************
          tnp=tend+1.d0
        endif
           if (tnp.gt.tend) then
c             ********************************
c             * 2a. ponding for whole period *
c             ********************************
           call pndinf(tstart,tend,rrfi,tp,tpp,fp,ntimes)
```

```
           ipond=1
        else
c             *******************************
c             * 2b. ponding ends at tnp      *
c             *******************************

c             ********************
c             *   ponded portion *
c             ********************
                call pndinf(tstart,tnp,rrfi,tp,tpp,fp,ntimes)
c             ********************
c             *  no pond portion *
c             ********************
                   call nopond(tnp,tend,rrfi,ntimes)

                  ipond=0
        endif
     endif
  100   continue

c**** infiltrate any water remaining in storage
     if (ipond.gt.0) then
        tstart = tend
        tend = 5000.d0 !should possibly change to 5000? llw
6.1.2022
           amtinf= bf(ntimes)+stor(ntimes)
        rrfi=0.d0
           call newtnp(tstart,tend,tnp,tp,tpp,rrfi,amtinf)
        tend=tnp
        call pndinf(tstart,tnp,rrfi,tp,tpp,fp,ntimes)
     endif

c******** write storm result table
     write(NUT,490)
     write(NUT,492)(sttime(i),endtim(i),rawrfi(i),i=1,nrain)
     write(NUT,495)
     write(NUT,500)
     write(NUT,504)
     ropeak=-10.d0
     rotpk=0.d0
     rfpeak=-10.d0
     rftpk=0.d0
      dro(1)=0.d0
     do 150 ii=1,ntimes
          if(ii.gt.1) dro(ii)=ro(ii)-ro(ii-1)
         if (ttp(ii).lt.9999.or.ttpp(ii).lt.9999) then
```

```
            write(NUT,502)
times(ii),ttp(ii),ttpp(ii),rint(ii),prec(ii),
     1            bf(ii),fpp(ii),f(ii),stor(ii),ro(ii),dro(ii)
          else
            write(NUT,503) times(ii),rint(ii),
     1
prec(ii),bf(ii),fpp(ii),f(ii),stor(ii),ro(ii),dro(ii)
         endif
c           ***************************************
c           * find peak ro and rrfi and times     *
c           ***************************************
         if (ropeak.lt.ro(ii)) then
            ropeak=ro(ii)/deltim
            rotpk = times(ii)
         endif
         if (rfpeak.lt.rint(ii)) then
            rfpeak=rint(ii)
            rftpk = times(ii)
         endif
  150 continue
      ROT = ro(ntimes) !cumulative runoff in cm
      Q = ROT*10.d0  !runoff in mm
      !print*,'q in gampt',q
      CINF = bf(ntimes)*10.d0 !cumulative infiltration, mm
     write(NUT,504)
     write(NUT,507)
     wbalck=bf(ntimes)+ro(ntimes)
     write(NUT,505)
CINF,CINF*AREA*10.D0,Q,Q*AREA*10.D0,wbalck*10.d0,
     & prec(ntimes)*10.d0
     write(NUT,506) ropeak*10.d0, rotpk, rfpeak*10.d0, rftpk
     write(NUT,504)
!    stop

c              *********************************************
  505   format(5x,'Event Statistics',/,
     1       8x,'Cum. Infiltration
=',f10.3,2x,'mm',2x,'=',2x,
     &       f10.3,2x,'m3',/,
     2       8x,'Runoff
=',f10.3,2x,'mm',2x,'=',2x,
     &       f10.3,2x,'m3',/,
     3       8x,'---------------------  ',10('-'),/,
     4       8x,'Cum. Inf+ Runoff       =',f10.3,2x,'mm',/,
     5       8x,'Total Rainfall         =',f10.3,2x,'mm',/)
  506   format(8x,'Peak Runoff Rate (mm/h)   =',f10.3,
     1              3x,'at time=',f10.3,' h',/,
```

```
     1         8x,'Peak Rainfall Int. (mm/h) =',f10.3,
     1              3x,'at time=',f10.3,' h',/)
  490   format(/,/,10x,28('-'),/,
     1             10x,'|',' Rainfall Distribution ',3x,'|',/,
     2             10x,'|',26('-'),'|',/,
     2             10x,'|',' Start',3x,'End
',6x,'RFi',3x,'|',/,
     3             10x,'|',' Time ',3x,'Time ',6x,'
',3x,'|',/,
     4             10x,'|',' -----',3x,'-----',6x,'---
',3x,'|',/,
     5             10x,'|','   h  ',3x,'  h
',5x,'cm/h',3x,'|',/,
     6             10x,'|',26('-'),'|')
  492   format(10x,'|',f6.2,3x,f6.2,3x,f6.2,2x,'|')
  495   format(10x,28('-'))
  500   format(/,/,10x,'Green-Ampt Test Routines',/,/,
     1         15x,'Based on work of Mein&Larson and Chu',/,
     2         1x,87('-'),/,
     3     3x,'Time',4x,'tp',4x,'tpp',5x,'R',7x,'P',7x,'F',7x,
     4        'fp',7x,'f',7x,'S',7x,'RO',7x,'dRO',/,
     5     3x,' h  ',4x,' h',4x,' h
',4x,'cm/h',5x,'cm',6x,'cm',5x,
     6        'cm/h',4x,'cm/h',5x,'cm',7x,'cm',7x,'cm')
  502   format(2x,f6.3,1x,f6.2,1x,f6.2,2x,f5.2,2x,f6.2,
     1
2x,f6.2,2x,f6.2,2x,f6.2,2x,f6.2,2x,f6.2,2x,f6.3,1x)
  503   format(2x,f6.3,2x,' npp ',2x,' npp ',2x,f5.2,2x,f6.2,
     1         2x,f6.2,2x,f6.2,2x,f6.2,2x,f6.2,2x,f6.2,2x,f6.2)
  504   format(1x,87('-'))
  507   format(3x,'Note:**** or 999 in tp or tpp means no
ponding in
     1period',/)
     end

     subroutine documnt
      IMPLICIT DOUBLE PRECISION (a-h, o-z)

     print *
     print *
     print *,'
*****************************************'
     print *,'           * Green-Ampt Unsteady Rainfall
*'
     print *,'           *   Version 0.8,  16/6/05,  jep, rmc
*'
```

```
     print *,'           *
*'
     print *,'           * To run type:
*'
     print *,'           *  gampt soilsfile rainfile (> outfile)
*'
     print *,'           *
*'
     print *,'           * where:
*'
     print *,'           *   soilsfile = soil input data
*'
     print *,'           *   rainfile  = rainfall inputs
*'
     print *,'           *   outfile   = output filename
*'
     print *,'           * OR
*'
     print *,'           *   Enter the Files Below
*'
     print *,'
*****************************************'
     print *
     print *

     return
     end

        subroutine outinp
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
     common /gampar/ vsatk, sav, wcsat, wcini, bm, deltim, stmax

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE

     write(NUT,100)
     write(NUT,102) vsatk,sav,wcsat,wcini,stmax,deltim

     return
c        1         2         3         4         5         6
7
c234567890123456789012345678901234567890123456789012345678901234
567890123456789
  100   format(/,/,10x,47('-'),/,
     1  10x,'|',3x,'Green-Ampt Solution for Unsteady
Rainfall',2x,'|',/,
     2  10x,'|',3x,' by J.E.Parsons, v0.8, R.Munoz-Carpena
',2x,'|',/,
```

```
     3  10x,'|',3x,' Routines from Papers by Mein and Larson
',2x,'|',/,
     4  10x,'|',3x,' 1971 and Chu, 1976. See Reference Sec.
',2x,'|',/,
     5  10x,'|',3x,' Version as of 16/6/05.  jep-rmc
',2x,'|',/,
     6  10x,47('-'),/)
  102   format(10x,46('-'),/,10x,'|',3x,'INPUT
PARAMETERS',25x,'|',/,
     1  10x,'|',2x,'Sat. K                =',f8.3,' cm/h
',2x,'|',/,
     2  10x,'|',2x,'Sav                   =',f8.2,' cm
',2x,'|',/,
     3  10x,'|',2x,'Sat. Water Content    =',f8.3,'
cm^3/cm^3',2x,'|',/,
     4  10x,'|',2x,'Initial Water Content =',f8.3,'
cm^3/cm^3',2x,'|',/,
     5  10x,'|',2x,'Maximum Surface Stor. =',f8.1,' cm
',2x,'|',/,
     6  10x,'|',2x,'Solution Time Step    =',f8.3,' h
',2x,'|',/,
     7  10x,46('-'),/)
     end


     subroutine nopond(tstart,tend,rrfi,ntimes)

      IMPLICIT DOUBLE PRECISION (a-h, o-z)
     common /gampar/ vsatk, sav, wcsat, wcini, bm, deltim, stmax
     common /gamp1/ tp,tpp,fp
     common /gamp2/ ttp(5000),ttpp(5000),fpp(5000)
     common /grunoff/ bf(5000),f(5000), stor(5000),
     1    ro(5000),prec(5000),rint(5000)
      common /deltaro/ dro(5000),times(5000)

c*********************
c* find the number of time steps...
c********************
       nsteps = int((tend-tstart)/deltim)
        !print*,'nsteps',nsteps
        !print*,'ntimes',ntimes
       do 50 kk=1,nsteps
         ntimes=ntimes+1
         times(ntimes)=times(ntimes-1)+deltim
         delinf = rrfi * deltim
         bf(ntimes)=bf(ntimes-1)+delinf
         fp=vsatk + (vsatk*bm*sav/bf(ntimes))
```

```
         f(ntimes)=rrfi
         prec(ntimes)=prec(ntimes-1)+delinf
         rint(ntimes)=rrfi
         fpp(ntimes)=fp
              ttp(ntimes)=tp
         ttpp(ntimes)=tpp
         stor(ntimes)=0.d0
         ro(ntimes)=ro(ntimes-1)
   50   continue
c*********************
c*  check that we really are at the end of the period
c*********************
     dterr = tend - times(ntimes)
     if (dterr.gt.0.d0) then
         ntimes=ntimes+1
         delinf = rrfi * dterr
         bf(ntimes)=bf(ntimes-1)+delinf
         fp=vsatk + (vsatk*bm*sav/bf(ntimes))
         f(ntimes)=rrfi
         prec(ntimes)=prec(ntimes-1)+delinf
         rint(ntimes)=rrfi
         fpp(ntimes)=fp
            ttp(ntimes)=tp
         ttpp(ntimes)=tpp
         stor(ntimes)=0.d0
         ro(ntimes)=ro(ntimes-1)
      endif
     times(ntimes)=tend
     return
     end

     subroutine pndinf(tstart,tend,rrfi,tp,tpp,fp,ntimes)

      IMPLICIT DOUBLE PRECISION (a-h, o-z)
      common /gampar/ vsatk, sav, wcsat, wcini, bm, deltim,
stmax
     common /gamp2/ ttp(5000),ttpp(5000),fpp(5000)
     common /grunoff/ bf(5000),f(5000), stor(5000),
     1    ro(5000),prec(5000),rint(5000)
      common /deltaro/ dro(5000),times(5000)

c**********************
c* find the number of time steps...
c*********************
       nsteps = int((tend-tstart)/deltim)
       do 50 kk=1,nsteps
         ntimes=ntimes+1
```

```
         times(ntimes)=times(ntimes-1)+deltim
         water= rrfi * deltim + stor(ntimes-1)
c            **********************
c            * make a guess for bigf
c            ***********************
             bbf = bf(ntimes-1)+water
         ctime=times(ntimes)
         call sschu(ctime,tp,tpp,bbf)
            delinf = bbf - bf(ntimes-1)
         bf(ntimes)= bbf
         fp=vsatk + (vsatk*bm*sav/bf(ntimes))
         f(ntimes)=fp
         if (water.gt.delinf) then
            stor(ntimes)=water-delinf
               if (stor(ntimes).gt.stmax) then
            ro(ntimes)=ro(ntimes-1)+(stor(ntimes)-stmax)
            stor(ntimes)=stmax
            else
            ro(ntimes)=ro(ntimes-1)
            endif
          else
            ro(ntimes)=ro(ntimes-1)
            stor(ntimes)=0.d0
          endif
             prec(ntimes)=prec(ntimes-1)+rrfi*deltim
          rint(ntimes)=rrfi
         fpp(ntimes)=fp
            ttp(ntimes)=tp
         ttpp(ntimes)=tpp
   50   continue
c*********************
c*  check that we really are at the end of the period
c*********************
     dterr = tend - times(ntimes)
     if (dterr.gt.0.d0) then
         ntimes=ntimes+1
         times(ntimes)=tend
         water= rrfi * dterr+ stor(ntimes-1)
c            **********************
c            * make a guess for bigf
c            ***********************
             bbf = bf(ntimes-1)+water
         ctime=times(ntimes)
         call sschu(ctime,tp,tpp,bbf)
            delinf = bbf - bf(ntimes-1)
         bf(ntimes)= bbf
         fp=vsatk + (vsatk*bm*sav/bf(ntimes))
```

```
         f(ntimes)=fp
         if (water.gt.delinf) then
            stor(ntimes)=water-delinf
               if (stor(ntimes).gt.stmax) then
            ro(ntimes)=ro(ntimes-1)+(stor(ntimes)-stmax)
            stor(ntimes)=stmax
            else
            ro(ntimes)=ro(ntimes-1)
            endif
          else
            ro(ntimes)=ro(ntimes-1)
            stor(ntimes)=0.d0
          endif
         prec(ntimes)=prec(ntimes-1)+rrfi*dterr
         rint(ntimes)=rrfi
         fpp(ntimes)=fp
            ttp(ntimes)=tp
         ttpp(ntimes)=tpp
      endif
     return
     end

     subroutine sschu(tim, tp, tpp, bff)

      IMPLICIT DOUBLE PRECISION (a-h, o-z)
     common /gampar/ vsatk, sav, wcsat, wcini, bm, deltim, stmax

c*****************
c* use Newton's method on chu's equation
c****************
c*
c****************
c*  set up problem
c****************
         iter = 0
         accpt=0.1d-04
         hh = bff - bm * sav * log (1.d0 + (bff/(bm*sav)))
     1           - vsatk * (tim-tp+tpp)

   10       iter=iter+1
         if (iter.gt.200) then
            write (*,12) iter, dhdf, hh, bff,error
   12   format(2x,'** it=',i4,' dhdf=',f8.4,' hh=',f8.4,
     1         ' bff =',f10.5,' error=',f10.6)
            stop
         endif
```

```
          dhdf = 1.d0 - ((bm * sav) /(bm*sav + bff))
          bbfnew = bff - hh/dhdf
          bff = bbfnew
          hh = bff - bm * sav * log (1.d0 + (bff/(bm*sav)))
     1           - vsatk * (tim-tp+tpp)
          error = abs(hh)
          if (error.gt.accpt) go to 10

      return
      end

      subroutine newtnp(tst,tend,tnp,tp,tpp,rrfi,amtinf)

       IMPLICIT DOUBLE PRECISION (a-h, o-z)
       common /gampar/ vsatk, sav, wcsat, wcini, bm, deltim,
stmax

      accpt=0.1d-04

      iter=0

      tnp = tend
      bftry = (tnp-tst)*rrfi + amtinf
      arglog = 1.d0 + bftry/(bm*sav)
      hnp = bftry - bm*sav*log(arglog)
      1    - vsatk*(tnp -tp +tpp)

   10   iter=iter+1
      tnpold=tnp
      if (iter.gt.200) then
c           write (*,12) iter, tst,dhnp, hnp, tnp,error
c   12   format(2x,'** it=',i4,' tst=',f8.4,' dhnp=',f8.4,'
hnp=',f8.4,
c     1          ' tnp =',f10.5,' error=',f10.6)
          tnp=tend+1.0
           return
      endif

      dhnp= rrfi - rrfi*(1.d0/arglog) - vsatk

      tnp = tnpold - hnp/dhnp

c** if tnp is negative  - fix added jep, 6/16/05
       if (tnp.lt.0.d0) tnp=0.d0

      bftry = (tnp-tst)*rrfi + amtinf
      arglog = 1.d0 + bftry/(bm*sav)
```

```
c    print *,"tnp,tst,rrfi,amtinf,bftry,bm,sav:",
c     1   tnp,tst,rrfi,amtinf,bftry,bm,sav
     hnp = bftry - bm*sav*log(arglog)
     1    - vsatk*(tnp -tp +tpp)
     error = abs(hnp)

     if (error.gt.accpt) go to 10

     return
     end
```

```
      SUBROUTINE cgampt(nrefga)
!C ----------------------------------------------------------------
---------------
! This subroutine removes singularity points from 5-minute
Green-Ampt results
! & calculates number of timesteps with excess runoff
C refga(5000,2): Matrix holding timestep (hr) and depth of
excess rainfall (mm)
C ----------------------------------------------------------------
---------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      common /gampar/ vsatk, sav, wcsat, wcini, bm, deltim,
stmax
      common /deltaro/ dro(5000),times(5000)
      common /cgam/ refga(5000,2)
      dimension dro1(5000),times1(5000)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE


C OUTPUT FILE
      !open(unit=22, file='gampout22.txt')
      write(NUT,495)
      write(NUT,500)
      write(NUT,495)

C INITIALIZE VARIABLES
      times1(1)=times(1)
      dro1(1)=dro(1)
      j=2
      tol=0.00010d0
      nrefga=0
      DO 10 i=2,5000
          deltim1 = times(i)-times(i-1)
          if (abs(deltim1-deltim).lt.tol) then
              times1(j)= times(i)
              dro1(j)=dro(i)
              j=j+1
          end if
10    continue
      DO 20 i=1,5000
          dro(i)=dro1(i)
          times(i)=times1(i)
          IF(dro(i).gt.0) THEN
            nrefga=nrefga+1
            refga(nrefga,1)=times(i)
```

```
            refga(nrefga,2)=dro(i)*10.d0
          END IF
          IF (times(i).gt.0) then
              write(NUT,502) times(i),dro(i)
          END IF
20    continue
C OUTPUT FORMATTING
495   format(1x,70('-'))
500   format(/,/,10x,'Green-Ampt 5-minute results',/,/,
     1         15x,'Based on work of Mein&Larson and Chu',/,
     2         1x,70('-'),/,
     3     3x,'Time',7x,'dRO',/,
     5     3x,' h  ',7x,'cm')
502   format(2x,f6.2,3x,f6.3,1x)

      RETURN
      END
```

```
      SUBROUTINE gapeak(Q,TC,qp,tp,Area)
C -----------------------------------------------------------------
--------------C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCC
C THIS SUBROUTINE PEAK FLOW AND TIME TO PEAK FLOW USING
TRIANGULAR METHOD    C
C INPUTS
C TC: time of concentration, passed in minutes from GATC and
converted to hours
C deltim: duration of excess rainfall to generate pulse, hours
C Q: Total runoff generated from Green-Ampt, mm
C tp: time to peak flow, hours
C qp: Peak flow, m3/s
C -----------------------------------------------------------------
---------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      common /gampar/ vsatk, sav, wcsat, wcini, bm, deltim,
stmax

C --  Convert time of concentration from minutes to hours
      TC=TC/60.d0
      !print*,'Q',q
      IF(Q.le.0) THEN
         qp=0.d0
        tp=qp
         RETURN
      END IF

C
C -- Calculate time of concentration and peak flow using NRCS
Triangular method
      tp=deltim/2.d0 + 0.6d0*TC
      qpmm=2.d0*Q/(2.67d0*tp) !qp in mm/hr
      qp = qpmm*Area*10.d0/3600.d0 !qp in m3/s
      !print*,'qp in gapeak',qp



      RETURN
      END
```

```
C ---------------------------------------------------------------
------------------
C  Program:   GREEN-AMPT PROGRAM
C ---------------------------------------------------------------
------------------
      SUBROUTINE GASH(P,DAYRO,DAYDN,DSM1,DSM2,DRECH,DBASEF,
     & DAYMO,DTHETA,DETA,SISTORE,BFloss,DSED,PEFF,DTHETA2,
     &  dstorvol,dmaxstor,drloss,dSIWater1,dminstor,DBF,DIRREFF)
C ---------------------------------------------------------------
------------------

C---------------------------------------------------------------
-----------------
c    Date      Modification
Initials
c   -------    ------------------------------                ---
---
c   5/25/2022 Initial program setup
llw
c
c    Date      Modification
Initials
c   -------    ------------------------------                ---
---
c   2/17/99    Check for 0.1<Ia/P<0.5
rmc
c   2/18/99    Added hyetograph output for 6 h storm
jep
c   2/18/99    ModIFy File Inputs for Erosion
jep
c   2/20/99    Roughed in MUSLE
jep
c   3/01/99    Checked erosion parameters and units
rmc
c   3/02/99    Additional work on Musle - units close
jep
c   3/03/99    Added hyetographs for storm types I & IA
rmc
c   3/05/99    output irs file for VFSMOD
jep
c   3/06/99    Input/Output files as in VFSMOD
jep
c   3/10/99    Checked Input/Output files as in VFSMOD
rmc
c   3/10/99    Cleanup - created hydrograph.f for
c              hydrograph subroutines, created io.f for
```

```
c             input and output related processing
jep
c   3/28/99   Erosion part: fixes in I30 calculation
c             after Chow and checked for consistency in
c             units, clean up; Hydro: added delay time
rmc
c   8/27/99   Added option to select dIFferent methods
c             for applying MUSLE, default is Foster,
c             2=Williams, 3=GLEAMS
c   10/01/99  Fixed array so that storm duration (D)
c             can now be up to 24h
rmc
c   10/26/99  implemented the project file concept as in vfsm
jep
c    3/09/00  Version changed to 0.9, general program cleanup
rmc
c   16/06/00  Version changed to 1.0, erosion output organized
rmc
c   16/03/02  Version changed to 1.06 to couple with VFSMOD,
c             author affiliation changed
rmc
c    4/18/03  Fixed K - computed IF we enter -1, other use
jep
c             entered value, also fixed dp output format
c    4/19/03  dp now being read in
jep
c    4/20/03  Runoff calculation for low CN revised
rmc
c    5/01/03  Added chacked for small runoff case to switch
rmc
c             to Williams sediment calculation that includes
c             runoff.
c   11/10/03  Reordered Erosion ieroty 1=Williams, 2=Gleams
c               3=Foster to coincide with changes in Shell
jep
c   11/13/03  Fixed coef. on Type Ia - did not add new
c               hyet curves
c   01/10/05  Added changes suggested by U. of Guelph group
rmc
c             v2.4.1
c   09/15/11  Rewritten hydrograph calculation using
convolution
c             of excess rain steps, v3.0.0
rmc
c   02/15/12  Added user table for 24-h hyetograph, v3.0.1
rmc
```

```
C --------------------------------------------------------------
-------------------
c Compiling for Win32 and Unix environments:
c       1. The i/o for these operating systems is dIFferent.
c       2. Change the comments in the finput.f program to
reflect
c          your operating system.   3/9/00
C --------------------------------------------------------------
-------------------
c COMMON/hydgph:
c       rot(208), runoff time (units)
c       roq(208), runoff rate (m3/s)
c       u(208,2), unit hydrograph
c COMMON/rain/:
c       rfix, maximum rain intensity (mm/h)
c       rti(200), rainfall time (hrs)
c       rfi(200), rainfall intensity (mm/h)
c       rcum(100,2), cumm rainfall (mm)
c       ref(100), excess rainfall intensity (mm/h)
c       ncum: number of steps IF user hyetograph is read
c other:
c       nref  = number of excess hyetograph steps
c       mref  = number of unit hydrograph steps
c       nhyet = number of hyetograph steps
c       vol(m3), volro (mm) = runoff volume
C --------------------------------------------------------------
------------------

C --------------------------------------------------------------
-----------------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
-----------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      DIMENSION
dstorvol(100),dmaxstor(100),dminstor(100),drloss(100),
     & dSIWater1(100)
      DIMENSION DAYRO(5555,100),DAYDN(5555,100),PEFF(5555,100),
     &
DTHETA(5555,100),DETA(5555,100),SISTORE(100),DBASEF(5555,100),
     &
DSED(5555,100),DRECH(5555,100),DSM1(5555,100),DSM2(5555,100),
```

```
     & DTHETA2(5555,100),DBF(5555,100),DAYMO(5555,100)
      DIMENSION AA(5555),SUMQO24(288),DIRREFF(5555,100)


C ----------------------------------------------------------------
------------------------
C  Get inputs and open files
C ----------------------------------------------------------------
------------------------
      CALL GAINPUTS(NA,Area,jstype,D,pL,Y,ITCTYPE,isoil,ek,
     C                  cfact,pfact,dp,ieroty,xIa,om,
     C
uFC,uWP,ZR,PFRAC,HM,dtheta,soilpt,Zstore)

      DO 50 K=1,100
        IF (INT(DPRECIP(1,K)).EQ.NZ2) THEN
            inode=K
        END IF
50    CONTINUE
      AA=0.D0
C ---Calculate volume of effective precipitation for Water
balance
      PEFF(JDAY,INODE)=P*AREA*10.D0
C ----------------------------------------------------------------
------------------------
c  Calculate storm hyetograph from SCS storm type
C ----------------------------------------------------------------
------------------------
       CALL hyetgh(jstype,P,D,xIa,
     &       ti,nref,a1,b1,bigE,raimax30,ndtime)
C ----------------------------------------------------------------
------------------------
C  Calculate runoff volume by Green-Ampt Method
C ----------------------------------------------------------------
------------------------
      CALL gampt(ndtime,D,Q,CINF,area)

      IF (Q.GT.0.D0) THEN
          CALL cgampt(nrefga)

C ----------------------------------------------------------------
------------------------
C  Calculate concentration time by large watershed methods
C ----------------------------------------------------------------
------------------------
         CALL gatc(ITCTYPE,Y,pL,Area,TC)
         !Dstep=0.24d0*tc
```

```
         !print*,times
C -------------------------------------------------------------
------------------------
C  Calculate peak flow and time to peak by NRCS triangular
method
C -------------------------------------------------------------
------------------------
         CALL gapeak(Q,TC,qp,tp,Area)
C -------------------------------------------------------------
------------------------
C  Generate unit hydrograph pulses using Haan's equation
C -------------------------------------------------------------
------------------------
         CALL unit_hyd(Q,Area,qp,tp,D,tc,mref)
       !print*,'q after unit hyd',q
C -------------------------------------------------------------
------------------------
C  Output hydrology results
C -------------------------------------------------------------
------------------------
!      CALL
results(P,CN,Q,Area,tc,xIa,jstype,D,pL,Y,qp,tp,qdepth,
c                  ieroty)

C -------------------------------------------------------------
------------------------
c  Calculate storm hydrograph
C -------------------------------------------------------------
------------------------
      CALL gatab_hyd(Area,mref,nrefga,ti,qp,tp,nhyd,NA,qcum,AA)
C -------------------------------------------------------------
------------------------
C DO the modified usle to get erosion stuff
C -------------------------------------------------------------
------------------------
      CALL musle(er,er1,erCoolm,ek,Y,pl,cfact,pfact,Area,Q,tc,P,
     C
D,isoil,dp,sconc,sconc1,sconc2,om,a1,b1,bigE,raimax30,qp,
     C      ieroty,sconc3)

C -------------------------------------------------------------
------------------------
C iF NO NEW RUNOFF OCCURS - add tail from previous day
C -------------------------------------------------------------
------------------------
      ELSE
          WRITE(NUT,1000)
```

```
          DO 770 I=1,5555
              AA(I)=AA(I)+STAIL(I,INODE)
770       CONTINUE
          CALL MWRITE(NA,AA)
      DO 780 I=1,5555-1
         STAIL(I,INODE)=0.D0
         IF (I.GT.288) THEN
             STAIL(I-288,INODE) = AA(I)
         END IF
780   CONTINUE
      if (AA(5555).ge.288) then
          stail(5555,INODE)=AA(5555)-288.d0
      else
          stail(5555,INODE)=0.d0
      end if
      END IF
c---------------------------------------------------------------
------------------------
C Use ThetaFAO to calculate initial soil moisture for next day
c --------------------------------------------------------------
------------------------
      IF (P.GT.0.0D0) THEN
          CINF=P-Q
      ELSE
          CINF=0.0D0
      END IF
      CALL thetafao(CINF,isoil,UFC,UWP,Zr,pfrac,Hm,
     & THETA,ETA,DPerc,inode,dtheta1,FC,WP,P,BFsm,DIRREFF)
C --------------------------------------------------------------
------------------------
C  Calculate deep percolation seepage/fractional redistribution
C --------------------------------------------------------------
------------------------
      !DBASEF(JDAY,INODE)=DBASEF(JDAY,INODE)-DBASEF(JDAY,JNODE)
      CALL DPSEEP(ISOIL,SISTORE,DPerc,INODE,BFloss,NA,AREA,
     &
soilpt,Zstore,RECHARGE,DSIwater,FC,WP,wcini,SWC2,DBASEF,DRECH,
     &
DSM2,dstorvol,dmaxstor,drloss,dSIWater1,dminstor,DBF,DTHETA2)

C --------------------------------------------------------------
------------------------
C  Calculate 24 hour flows
C --------------------------------------------------------------
------------------------
      SUMQO24(1)=0.5D0*AA(1)*5.D0*60.D0*0.0283168D0
      DO 11 I=2,288
```

```
          SUMQO24(I)=SUMQO24(I-1)+0.5d0*(AA(I-1)+
     &        AA(I))*5.d0*60.d0*0.0283168d0 !Calculating the sum
by taking average of timesteps, multiplying over time step to
get volume, and converting to cubic meters
11    CONTINUE

!
!C ----------------------------------------------------------------
-------------------------
!C  Set up hydrograph recession coeffs
!C ----------------------------------------------------------------
-------------------------
!      dailyq=SUMQO24(288)
!      arec=0.00104d0
!      brec=1.520d0
C -----------------------------------------------------------------
------------------------
C  Write runoff to STORAGE file
C -----------------------------------------------------------------
------------------------


      DAYRO(JDAY,inode)=SUMQO24(288)
      DAYDN(JDAY,Inode)=CINF*Area*10.d0
      DAYMO(JDAY,INODE)=Q*AREA*10.D0-SUMQO24(288)
      DTHETA(JDAY,INODE)=THETA
      !DTHETA2(JDAY,INODE)=SWC2
      DETA(JDAY,INODE)=ETA*Area*10.d0
      !DRECH(JDAY,INODE)=RECHARGE
      DSM1(JDAY,INODE)=dtheta1*Area*Zr*10000.d0
      !DSM2(JDAY,INODE)=DSIwater



!DBASEF(JDAY,INODE)=BFsm*Area*Zr*10000.d0+DBASEF(JDAY,INODE)
      IF (Q.GT.0.D0) THEN
          IF (ieroty.eq.1)THEN !!     1) Williams (1975)
              DSED(JDAY,INODE)=SCONC1
          ELSEIF (ieroty.eq.2)THEN !! 2) GLEAMS / daily CREAMS
              DSED(JDAY,INODE)=SCONC2
          ELSEIF (ieroty.eq.3)THEN !! 3) Foster et al. (1977)
              DSED(JDAY,INODE)=SCONC
          ELSEIF (ieroty.eq.4)THEN !! 4) Cooley (1980) - Design
Storm
              DSED(JDAY,INODE)=SCONC3
          ENDIF
      ELSE
```

```
            DSED(JDAY,INODE)=0.D0
        END IF

        !L=1
        !    DO WHILE (dailyq.gt.0.d0.and.L.le.213)
        !        dqdt=arec*dailyq**brec
        !        IF (dqdt.ge.dailyq) then
        !            dqdt=dailyq
        !        END IF
        !        DBASEF(JDAY+L,INODE)=dailyq-dqdt +
DBASEF(JDAY+L,INODE)
        !        dailyq=dailyq-dqdt
        !        L=L+1
        !END DO
C ----------------------------------------------------------------
-----------------
C    OUTPUT - FORMAT
C ----------------------------------------------------------------
-----------------
1000  FORMAT(6X,'NO NEW FLOW GENERATED ON THIS DAY')

      RETURN
      END SUBROUTINE
```

```
C ----------------------------------------------------------------
---------------------
      SUBROUTINE
gatab_hyd(Area,mref,nrefga,ti,qp,tp,nhyd,NA,qcum,AA)
C ----------------------------------------------------------------
---------------------
C   Calculation of hydrograph by convolution (Chow, 1987) of SCS
unit
C   hydrograph and excess hyetograph
C ----------------------------------------------------------------
---------------------
C      version 3.0.1, Last Modified: See Modifications below
C      WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C      Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons,
BAE (jep)
C                  University of Florida        BAE, NC State
University
C                  Gainesville, FL 32611        Raleigh, NC
27695-7625(USA)
C                  e-mail: carpena@ufl.edu
C ----------------------------------------------------------------
---------------------
C   DECLARE VARIABLES
C   DECLARE VARIABLES
C---Inputs
C Q: runoff volume, mm
C Area: Watershed area, ha
C mref: number of unit hydrograph steps
C nrefga: number of excess (effective) hyetograph steps from
green-ampt
C ti: Initial time when runoff is generated, hr
C qp: Peak flow, m3/s
C tp: Time to peak flow, hr
C nhyd: Number of timesteps in final convolution hydrograph
C Dstep:Timestep between flow calculations
C NA: Stream number
C---Other variables
C TIME1:  Time for ss of results (hrs)
C TIME2:  Time for End of results (hrs)
C cqdepth5: cumulative flow (mm/hr) of unit hydrograph
C dt5: time step (hr)
C u(5000,2):matrix holding time (t5) in column 1 and  unit
hydrograph flow (qi5, m3/s) in column
C qh(5000,3): matrix holding time (hr) in col 1, hydrograph
(m3/s) in col 2, and accumulated runoff (m3) in col 3
C unitq: volume of flow in unit hydrograph (mm)
C def: Same as dt5, time step (hr)
```

```
C A:
C qcum: cumulative runoff volume (m3/day)
C ref(5000,2): Matrix holding timestep (hr) and depth of excess
rainfall (mm)
C qpdepth: Qpeak in mm
c qhstep(5000,3): Matrix holding time (hr) in col 1, runoff
convolution hydrograph (m3/s) in col 2, and accumulated runoff
(m3) in col 3
C qh(5000,3) and qhstep(5000,3) are essentiall the same as each
other, but qhstep is set at a user defined timestep
C H(i): flow at each timestep (ft3/s)
C -------------------------------------------------------------
---------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)
      COMMON/hydgph/u(5000,2),qh(5000,3)
      DIMENSION H(5555),AA(5555)
      !DATA H/5555*0.d0/
      common /cgam/ refga(5000,2)
!c------INITIALIZATIONS TO REMOVE IN CUENCA
!      STAIL=0.D0
!      AA=0.D0
C--------------------------------------------------------------
---------------------
C     FIND INODE NUMBER
C--------------------------------------------------------------
--------------------
      DO 10 J=1,100
         IF (INT(DPRECIP(1,J)).EQ.NZ2) INODE=J
10    CONTINUE
C -------------------------------------------------------------
---------------------
      UNIT=5.D0
      IF (nrefga.gt.0) then
       WRITE(NUT,205)mref,nrefga
       cqdepth5=0.d0
       DO 40 i=1,mref
           cqdepth5=u(i,2)*360.d0/Area+cqdepth5
40    CONTINUE
       dt5=u(2,1)-u(1,1)
       unitq=cqdepth5*dt5
       DO 50 i=1,nrefga
c        WRITE(*,'(2f10.4)')(ref(i,j),j=1,2)
```

```
c        WRITE(NUT,'(2f10.4)')(ref(i,j),j=1,2)
50    CONTINUE

C ---Apply convolution of the u and ref values to obtain
hydrograph
       Def=u(2,1)-u(1,1)
       qp=0.d0

       qcum=0.d0
        H=0.D0
       DO 70 k=1,nrefga+mref-1
           qh(k,1)=refga(1,1)+(k-1)*Def
           qh(k,2)=0.d0
           qh(k,3)=0.d0
           DO 60 i=1,k
               qh(k,2)=qh(k,2)+refga(i,2)*u(k-i+1,2)
60       CONTINUE
            !write(22,*)k,refga(k,2)
            !write(22,*)k,u(k,1)
             !write(22,*)k,u(k,2)
            !write(22,*)k,qh(k,3)
           IF(qh(k,2).gt.qp) tp=qh(k,1)
               qp=dmax1(qh(k,2),qp)
          !END IF
70    CONTINUE
       qpdepth=qp*360.d0/Area
c-rmc- need to swift position of hydrograph within h so first
value correspond to no. of steps
c----- for ponding, ini (begining of hydrograph)
c      INI=INT(qh(1,1)*60.d0/5.d0)
c      print*,'INI=',INI
        IF (K.GT.5555) THEN
            K=5555
       END IF
       DO 80 i=1,k-1
c -rmc- removed output of hydrographs in SI units, so only the
English units graph is shown
c------(like in unith)
c
WRITE(NUT,'(3f10.4)')(qh(i,j),j=1,2),qh(i,2)*360.d0/Area
c         H(i+INI-1)=qh(i,2)/(0.3048d0**3.d0)
           H(i)=qh(i,2)/(0.3048d0**3.d0)
           qcum=qcum+qh(i,2)*(3600.d0*5.d0/60.d0)
80    CONTINUE
       TIME1=qh(1,1)
        !print*,'time1',time1
       TIME2=qh(i-1,1)
```

```
       qh(1,3)=0.d0
       DO 61 i=2,k-1
           qh(i,3)=qh(i-1,3)+qh(i,2)*3600*Def !MAC 04/10/12
Accumulated (m^3)
61     CONTINUE

C -----------------------------------------------------------------
---------------------
       nhyd=k-1
        tpi=TIME1
       WRITE(NUT,950)tpi
       WRITE(NUT,1000)qp,qpdepth,qp/(0.3048d0**3.d0)
       WRITE(NUT,1100)tp,tp*60.d0
       WRITE(NUT,1200)nhyd


C -----------------------------------------------------------------
-------------------------
C Pass hydrograph to STREAM flow MATRIX SS
C -----------------------------------------------------------------
-------------------------
       H(5555)=nrefga+mref-1
       INTERV=nrefga+mref-1
       IF(INTERV.GT.440) INTERV=440
       ITIME1=int(60.d0*TIME1/UNIT)
       ITIME2=int(60.d0*TIME2/UNIT)
       !CALL MREAD(NA,AA)
       NUMX=INT(UNIT/5.d0+.01d0)
C      DO 750 I=1,INTERV
C         AA(I)=AA(I)+H(I)
C750   CONTINUE
       ICOUNT=1
!      DO 750 I=ITIME1,ITIME2 !LW 11.14.2022
        DO 750 I=ITIME1,5555-1
           AA(I)=AA(I)+H(ICOUNT)
           !print*,'hi,hicount',H(I),H(Icount)
           !H(ICOUNT)=H(ICOUNT)+STAIL(I,INODE)
           ICOUNT=ICOUNT+1
750   CONTINUE
       DO 770 I=1,5555-1
           AA(I)=AA(I)+STAIL(I,INODE)
770   CONTINUE
       AA(5555)=INTERV*NUMX
        !print*,'num in gatabhyd',AA(5555)
       CALL MWRITE(NA,AA)
C -----------------------------------------------------------------
-------------------------
```

```
C Print hydrograph (units in CFS and AF)
C --------------------------------------------------------------
-------------------------
       KTYPE=0
        !print*,'qcum in tabhyd',qcum
        !print*,'time1',Time1
       XMAX=qp/(0.3048d0**3.d0)
       SUM=qcum/1233.48d0 !converts qcum from m3/d to ac-ft/day
       CALL OASB(KTYPE,H,INTERV,XMAX,UNIT,SUM,TIME1,TIME2)
C --------------------------------------------------------------
-------------------------
C Pass tail along for next day
C --------------------------------------------------------------
-------------------------
      DO 785 I=1,5555-1
         STAIL(I,INODE)=0.D0
         IF (I.GT.288) THEN
             STAIL(I-288,INODE) = AA(I)
         END IF
785   CONTINUE
      if (AA(5555).ge.288) then
          stail(5555,INODE)=AA(5555)-288.d0
      else
          stail(5555,INODE)=0.d0
      end if
C --------------------------------------------------------------
-------------------------
C Output when there is no new runoff
C --------------------------------------------------------------
-------------------------
      ELSE
        WRITE(NUT,1300)
c        ----------
        H(5555)=0.d0
        DO 780 i=1,5555-1
            H(i)=0.d0
780     CONTINUE
       INTERV=288
       UNIT=5.d0
       ITIME1=1
       ITIME2=288
       CALL MREAD(NA,AA)
       NUMX=INT(UNIT/5.d0+.01d0)
       ICOUNT=1
       DO 790 I=ITIME1,ITIME2
           AA(I)=AA(I)+H(ICOUNT)
           H(ICOUNT)=H(ICOUNT)+STAIL(I,INODE)
```

```
           ICOUNT=ICOUNT+1
790   CONTINUE
       DO 795 I=1,665
           AA(I)=AA(I)+STAIL(I,INODE)
795   CONTINUE
       AA(5555)=INTERV*NUMX
       CALL MWRITE(NA,AA)
C -------------------------------------------------------------
-------------------------
C Print hydrograph (units in CFS and AF)
C -------------------------------------------------------------
-------------------------
       !KTYPE=0
       !!XMAX=qp/(0.3048d0**3.d0)
       !!SUM=qcum/1233.48d0 !converts qcum from m3/d to ac-
ft/day
   !     XMAX = 0.D0
   !     SUM = 0.D0
   !     TIME1=0.D0
   !     TIME2=24.D0
       !CALL OASB(KTYPE,H,INTERV,XMAX,UNIT,SUM,TIME1,TIME2)
      END IF
C -------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C -------------------------------------------------------------
----------
c-rmc- change to stop displaying the hydrograph in SI units
(like in unith)
c205   FORMAT(4X,'Number of unit hydrograph steps (n)  = ',i5,/,
c     1 4X,'Number of excess hyetograph steps (m) =',i5,/,/,
c     2 4X,'b) Final Hydrograph (CONVOLUTION):',/,
c     3 4X,'Time(h)   q(m3/s)   q(mm/h)',/,3X,30('-'))
205   FORMAT(4X,'Number of unit hydrograph steps (n)  = ',i5,/,
     1 4X,'Number of excess hyetograph steps (m) =',i5,/,
     2 4X,30('-'),/,4X,'b) Final Hydrograph (CONVOLUTION):')
950   FORMAT(4X,'Time to Ponding =',f8.3,' hr')
1000  FORMAT(4X,'Peak flow    =',f9.3,' m3/s = ',f9.4,' mm/h =
',f9.4,
     & ' cfs')
1100  FORMAT(4X,'Time to peak =',f8.2,' h    = ',f8.2,' min')
1200  FORMAT(4X,'Number of final hydrograph steps (nhyd) =
',i5,/)
1300  FORMAT(/,/,4X,'No new runoff generated on this day.'/)

      RETURN
      END SUBROUTINE gatab_hyd
```

```
C --------------------------------------------------------------
--------------
!     TIMEC - SUBROUTINE TO CALCULATE TIME OF CONCENTRATION
C --------------------------------------------------------------
--------------
      SUBROUTINE gatc(ITCTYPE,Y,pL,Area,TC)
C --------------------------------------------------------------
--------------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCC
C THIS SUBROUTINE CALCULATES TIME OF CONCENTRATION USING SEVERAL
METHODS     C
C INPUTS:
C ITCTYPE: METHOD FLAG FOR CALCULATING TC [0-4]
C             WHERE 0 = DEFAULT (AVERAGE OF ALL METHODS MINUS
MINIMUM VALUE, PREFERRED)
C                   1 = WILLIAMS METHOD (TC1)
C                   2 = JOHNSTONE-CROSS METHOD (TC2)
C                   3 = BRANSBY-WILLIAMS METHOD
C                   4 = PASSINI METHOD
C Y: Slope of source area, m/m                              C
C pL:     Channel length - Length of longest watercourse, m
C Area : AREA, HA
C --------------------------------------------------------------
---------
C VARIABLES:
C
C XB: BASIN LENGTH, MILES
C
C XF: LENGTH OF LONGEST WATERCOURSE, FT
C
C XK: LENGTH OF LONGEST WATERCOURSE, KILOMETERS               C
C DCIRC = DIAMETER(MI) OF A CIRCULAR BASIN OF AREA
C AK = AREA, SQUARE KILOMETERS
C AM = AREA, SQUARE MILES
C S = BASIN SLOPE, %
C SFPM = BASIN SLOPE, FEET/MILES
C SMK = AVERAGE SLOPE, METERS/KILOMETERS
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCC
C --------------------------------------------------------------
--------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
--------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
```

```
C      PARAMETER (5555=INT(600))
      DIMENSION TCI(5)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
! DIMENSION A(5555),B(5555)
! COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
!& SNODE(5555,100),STAIL(5555,100)
! COMMON/INPUTS/DATAINP(100,100)
! COMMON/NINOUT/NUT,nut,NIPR,NSSS,NZ2
C -------------------------------------------------------------
--------------
C   INITIALIZE VARIABLES
C -------------------------------------------------------------
--------------
!      TIME=0.d0

C -------------------------------------------------------------
--------------
C   CONVERT INPUTS TO VARIABLES
C -------------------------------------------------------------
--------------
      XF=pL*3.28084d0
      XB=XF/5280.d0
      XK = pL/1000.d0
      PI=ACOS(-1.D0)
      AM = Area*0.00386102D0
      DCIRC = SQRT(4.d0*AM/PI)
      AK = Area*.01d0
      S=Y*100.d0
      SFPM = Y*5280.d0
      SMK = Y*1000.d0

C -------------------------------------------------------------
--------------
C WRITE OUTPUT HEADER
C -------------------------------------------------------------
--------------
      !WRITE(NUT,901)NA,NB
C -------------------------------------------------------------
--------------
C   WILLIAMS METHOD - TC(1)
C   ---------------
C   VARIABLES
C XB = BASIN LENGTH, MILES
C AM = BASIN AREA, SQUARE MILES
C DCIRC = DIAMETER(MI) OF A CIRCULAR BASIN OF AREA
C S = BASIN SLOPE, %
```

```
C --------------------------------------------------------------
--------------
      TCI(1) = 60.D0*XB*(AM**0.4D0)*(DCIRC**(-1.D0))*(S**(-
0.2D0))
C --------------------------------------------------------------
--------------
C   JOHNSTONE-CROSS METHOD - TC(2)
C   ---------------
C   VARIABLES
C XB = BASIN LENGTH, MILES
C SFPM = BASIN SLOPE, FEET/MILES
C --------------------------------------------------------------
--------------
      TCI(2) = 300.D0*(XB**0.5d0)*(SFPM**(-0.5d0))
C --------------------------------------------------------------
--------------
C   BRANSBY-WILLIAMS METHOD - TC(3)
C   ---------------
C   VARIABLES
C XK = MAINSTREAM LENGTH, KILOMETERS
C AK = CATCHMENT AREA, SQUARE KILOMETERS
C SMK = AVERAGE SLOPE, METERS/KILOMETERS
C --------------------------------------------------------------
--------------
      TCI(3) = 58.5d0*XK*(AK**(-0.1d0))*(SMK**(-0.2d0))
C --------------------------------------------------------------
--------------
C   PASSINI METHOD - TC(4)
C   ---------------
C   VARIABLES
C AK = BASIN AREA, SQUARE KILOMETERS
C XK = LENGTH OF MAIN CHANNEL, KM
C Y = AVERAGE SLOPE OF BASIN, M/M
C --------------------------------------------------------------
--------------
      TCI(4) = 6.48d0*((AK*XK)**0.333d0)*(Y**(-0.5d0))
C ----------------------------------------------
C AVERAGE OF ALL METHODS - TC(5)
C ----------------------------------------------
      TCI(5) = (TCI(1)+TCI(2)+TCI(3)+TCI(4))/4.d0
      IMIN = MINVAL(TCI) !Identify minimum calculated TC
      TCI(5) = (TCI(1)+TCI(2)+TCI(3)+TCI(4)-IMIN)/3.d0
!Recalculate average using 3 highest values
C ----------------------------------------------
C DETERMINATION OF TIME OF CONCENTRATION
C TCF = TIME OF CONCENTRATION USED FOR CALCULATIONS
C ----------------------------------------------
```

```
      IF (ITCTYPE.EQ.0) THEN
          TC=TCI(5)
      ELSE IF (ITCTYPE.EQ.1) THEN
          TC=TCI(1)
      ELSE IF (ITCTYPE.EQ.2) THEN
          TC=TCI(2)
      ELSE IF (ITCTYPE.EQ.3) THEN
          TC=TCI(3)
      ELSE IF (ITCTYPE.EQ.4) THEN
          TC=TCI(4)
      END IF
C ----------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C ----------------------------------------------------------------
----------
      WRITE(NUT,900)
      WRITE(NUT,901)Y,pL,Area,ITCTYPE
      WRITE(NUT,921)TCI(5),TCI(1),TCI(2),TCI(3),TCI(4)
      WRITE(NUT,930)
900   FORMAT(/,1x,76('-'),/,4x,'MODEL TIME OF CONCENTRATION',/,
     C  1x,76('-'))
901   FORMAT(
     C /,7X,'Y - Slope of source area, m/m:',f5.2
     C ,/,7X,'pL - Length of longest watercourse, m:',f7.1
     C ,/,7X,'A -  AREA, HA:',f8.1
     C ,/,7X,'CHOSEN TC CALCULATION METHOD:',I1)
921   FORMAT(
     C 7X,'RESULTS OF EACH TC CALCULATION METHOD, MINUTES'
     C ,/,7X,'DEFAULT METHOD INCLUDES AVERAGE OF ALL METHODS
     & MINUS MINIMUM'
     C ,/,10X,'0: DEFAULT                       =',F5.2
     C ,/,10X,'1: WILLIAMS METHOD (TC1)         =',F5.2
     C ,/,10X,'2: JOHNSTONE-CROSS METHOD (TC2)   =',F5.2
     C ,/,10X,'3: BRANSBY-WILLIAMS METHOD        =',F5.2
     C ,/,10X,'4: PASSINI METHOD                 =',F5.2)
930   FORMAT(/,1x,76('='),/)


      RETURN
      END
```

```
     SUBROUTINE
getinp(NA,CN,Area,jstype,D,pL,Y,ek,cfact,pfact,isoil,
     C            ieroty,dp,om,uFC,uWP,ZR,PFRAC,HM,
     C            dtheta,soilpt,Zstore,inodeloc,arec,brec)
C --------------------------------------------------------------
------------------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
------------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)
      COMMON/INPUTS/DATAINP(100,100)
      common /gampar/ vsatk,sav,wcsat,wcini,bm,deltim,stmax

COMMON/rain/rfix,rti(5000),rfi(5000),rcum(5000,2),ref(5000,2),nc
um
      dimension dtheta(5555,100)
c      CHARACTER*20 isoil
C --------------------------------------------------------------
------------------------
      !READ(nut,*)NA,CN,Area,jstype,D,pL,Y
      DO 20 J=1,100
          IF (INT(DPRECIP(1,J)).EQ.NZ2) THEN
              INODE=J
          END IF
20    CONTINUE

      NA=INT(DATAINP(JCOUNT,4))
      CN=DATAINP(JCOUNT,5)
      Area=DATAINP(JCOUNT,6)
      jstype=INT(DATAINP(JCOUNT,7))
      D=DATAINP(JCOUNT,8)
      pL=DATAINP(JCOUNT,9)
      Y=DATAINP(JCOUNT,10)
C --------------------------------------------------------------
------------------------
c READ INPUTS: Soil Erosion Calculations:
C --------------------------------------------------------------
------------------------
c   isoil = soil type (Character), see musle.f data for list
soil types
c   ek     = soil erodibility
c   cfact  = C factor
```

```
c   pfact  = P factor
c   dp  = sediment size (d50) in cm. If dp= -1 dp is set based
on "isoil
C -----------------------------------------------------------------
------------------------
      !READ(nut,'(A)')isoil
      !READ(nut,*) ek, cfact, pfact, dp
      isoil=DATAINP(JCOUNT,11)
      ek=DATAINP(JCOUNT,12)
      cfact=DATAINP(JCOUNT,13)
      pfact=DATAINP(JCOUNT,14)
      dp=DATAINP(JCOUNT,15)
C -----------------------------------------------------------------
------------------------
c     Convert dp to um
C -----------------------------------------------------------------
------------------------
      dp=dp*10000.d0
C -----------------------------------------------------------------
------------------------
c-  ieroty = select method to estimate storm erosion:
c-     0 or not present = Foster's method for R-factor
c-     1 = Using Williams R-factor
c-     2 = Using R-factor from GLEAMS with daily rainfall
C -----------------------------------------------------------------
------------------------
      !READ(nut,*,END=22) ieroty
      ieroty=INT(DATAINP(JCOUNT,16))
      IF ((ieroty.lt.3).and.(ieroty.ge.0)) GO TO 24
22       ieroty=1
24    CONTINUE
C -----------------------------------------------------------------
------------------------
c om = % soil organic matter, read IF ek <0
C -----------------------------------------------------------------
------------------------
      om = 2.0d0
      IF (ek.lt.0.d0) THEN
         !READ(nut,*,END=32)om
          om=DATAINP(JCOUNT,17)
         END IF
c------------------------------------------------------------------
------------------------
c Read inputs for THETAFAO process
c Wcini:
c uFC:
c uWP:
```

```
c Zr:
c Pfrac:
c Hm:
c soilpt: soil porosity (m3/m3), if equal to 0 then will be
determined based on soil texture
c Zsoil: Depth between top surface elevation and riverbed (or
depth of soil horizon) (m)
c ZStore: Depth of intermediatee storage (depth of soil horizon
- rooting depth) (m)
c---------------------------------------------------------------
-----------------------
      IF (JDAY.EQ.1) THEN
          WCINI=DATAINP(JCOUNT,18)
      ELSE
          WCINI = DTHETA(JDAY-1,INODE)
      END IF
      uFC=DATAINP(JCOUNT,19)
      uWP=DATAINP(JCOUNT,20)
      ZR=DATAINP(JCOUNT,21)
      PFRAC=DATAINP(JCOUNT,22)
      HM=DATAINP(JCOUNT,23)
      soilpt = DATAINP(JCOUNT,24)
      Zsoil = DATAINP(JCOUNT,25)
      Zstore = Zsoil-ZR

C ---- Inputs for runoff recession only if node is first
upstream contributor for watershed
      INODELOC = DATAINP(JCOUNT,26)
      AREC=DATAINP(JCOUNT,27)
      BREC=DATAINP(JCOUNT,28)
C --------------------------------------------------------------
-----------------------
32    CONTINUE
      IF(jstype.gt.4) THEN
c-- For user defined case (jstype=4), read "tmid"(h)first and
then 24-h P/P24 curve
c-- "tmid" is stored in first position of the rcum(i,j) array
         READ(nut,*,END=40)rcum(1,1)  !! form where is is
obtaining this number ??
         PRINT*,rcum(1,1)
         IF (rcum(1,1).eq.0) THEN
         PRINT*,'ERROR: the first value must be tmid(h),
followed'
     &      ,'in the next lines by the cumulative rainfall  that
must'
     &      ,'begin with (0,0) and end with (24,1)'
         STOP
```

```
       END IF
       rcum(1,2)=0.5d0
       DO 35 i=2,5000
          READ(nut,*,END=40)(rcum(i,j),j=1,2)
          PRINT*,i,(rcum(i,j),j=1,2)
          IF(rcum(i,1).eq.24) GO TO 40
35     CONTINUE
40     ncum=i
       IF ((rcum(ncum,1).ne.24).or.(rcum(ncum,2).ne.1)) THEN
          PRINT*,'ERROR: the cumulative rainfall must begin with
',
     &       '(0,0) and end with (24,1)',i
          STOP
       END IF
       END IF
      RETURN
      END SUBROUTINE getinp
```

```
C ----------------------------------------------------------------
---------
      SUBROUTINE hydrog(DAYQI,DAYQO)
C ----------------------------------------------------------------
---------
!This subrouting is to address with future DATA from
hydroelectric Station
!bellow Sandillal Dam.
!This station just only works at nights, so maybe the DATA will
be: StarTime
!StopTime and Discharge (I guess it will be constant)
C ----------------------------------------------------------------
---------
C   DECLARE VARIABLES
C ----------------------------------------------------------------
---------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)
      COMMON/INPUTS/DATAINP(100,100)
      DIMENSION
A(5555),DAYQI(5555,100),DAYQO(5555,100),HYDRO(5555,100)
C ----------------------------------------------------------------
---------
      SS=SS1
      NA=INT(DATAINP(JCOUNT,4))
      FTIME=DATAINP(JCOUNT,5)
      ETIME=DATAINP(JCOUNT,6)
      DIS=DATAINP(JCOUNT,7)
      !DISANG=DATAINP(JCOUNT,8)
      !DISANG=DIS/(0.3048d0**3) !Convertion to CFS
C ----------------------------------------------------------------
---------
C   CONVERSION
C ----------------------------------------------------------------
---------
      DISANG=DIS/(0.3048d0**3.D0) !Conversion to CFS !
      TIME=0.d0
      NNUMBER=0 !should we add INT (...) to this value 10.24.18
      DO 50 I=1,5555-1
         TIME=TIME+0.08333d0
         Hydro(I,1)=TIME
         IF (TIME.GT.FTIME.AND.TIME.LE.ETIME) THEN
            SS1(I,NA)=SS1(I,NA)+DISANG
            Hydro(I,2)=SS1(I,NA)*(0.3048d0**3.D0) !Return
hydrograph in CMS
```

```
            ELSE
               Hydro(I,2)=SS1(I,NA)*(0.3048d0**3.D0)
            END IF
            IF (SS1(I,NA).GT.0) THEN
               NNUMBER=NNUMBER+1
            END IF
 50      CONTINUE
        SS1(5555,NA)=NNUMBER

         RETURN
        END SUBROUTINE hydrog
```

```
C --------------------------------------------------------------
---------------
C  Disaggregation Daily Precipitation
C --------------------------------------------------------------
---------------
      SUBROUTINE hyetgh(jstype,P,D,xIa,
     &       ti,nref,a1,b1,bigE,raimax30,ndtime)
C --------------------------------------------------------------
---------------
c      version 3.0.1, Last ModIFied: See ModIFications below
C      WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C      Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons,
BAE (jep)
C                  University of Florida        BAE, NC State
University
C                  Gainesville, FL 32611        Raleigh, NC
27695-7625(USA)
C                  e-mail: carpena@ufl.edu
C --------------------------------------------------------------
---------------
c ! Where P'(t)= is the cumulative hyetograph for the given
duration
c !       Pd is the total rainfall for the given period (mm)
c !       D is the storm duration in hours
c !       t is current time from start of storm in hours
c !
c ! storm type II or III - from Haan
c ! hyetograph for 24 hour storms
c !
c ! P(t)        T (  24.04  ) ^ 0.75
c ! ---- = 0.5+---(---------)
c ! P24        24 (2|T|+0.04)
c !
c !  where T=t -12  (with t in hours)
c !        P24 is the 24h storm
c !
c ! storm type I  - fitted from SCS tabular data - rmc 03/04/99
c !
c ! hyetograph for 24 hour storms
c !
c !                    (     -0.1617    ) ^ 0.5853
c !       | 0.4511+ T (----------------)       ;for [-
3.0163|T|+0.013]<0
c ! P(t)  |           (-3.0163|T|+0.013)
c ! ---- =|
c ! P24   |
```

```
c !       |   0.5129                          ;for [-
3.0163|T|+0.013]>0
c !
c !  where T=t -9.995  (with t in hours)
c !
c ! storm type IA  - fitted from SCS tabular data - rmc 03/04/99
c !
c ! hyetograph for 24 hour storms
c !
c ! P(t)              (     0.0843     ) ^ 0.4228
c ! ---- = 0.3919+ T (----------------)
c ! P24               (120.39|T|+0.3567)
c !
c !  where T=t -7.96  (with t in hours)
c !
c ! For any storm of any duration (from Haan et. al.(1994), eq.
3.7)
c !
c ! P'(t)    P(tmid+t-D/2) - P(tmid-D/2)
c ! ----- = -------------------------
c ! Pd       P(tmid+D/2) - P(tmid-D/2)
c !
c ! where where tmid=12. Alternatively (Munoz-Carpena and
Parsons,2004),
c ! tmid=12.00 for storm type II & III, tmid=9.995 for storm
type I, and
c ! tmid=7.960 for storm type IA
c !
c-------------------------------------------------------------
-------
c Variable Definitions
c-------------------------------------------------------------
-------
!   Ndtime: number of timesteps based on storm duration and
timestep length
!    Tmid, a1, b1: scaling factors for scs storm types
!    Tminus, tplus: scaling factors of time midpoint plus or
minus ½ storm duration
!    PM, PP: scaling factors
!    Rti(i): array holding start time (actual time) of each time
step (starts at 0)
!    Tsmall: at each time step, tmid (scaling factor based on
storm type) plus current actual time minus ½ storm duration
!    Ptp: ScStorm(jstype, tsmall)
!    Pd: precipitation, mm
!    Cumtotal: cumulative total rainfall for a given storm of
volume and duration
```

```
!    Ti: (change in time/cumulative total rainfall) * initial
abstraction + previous time
!    i.e. time that runoff starts to be generated ** exists to
be passed to tab_hyd
!    Pcumtot: cumulative rainfall at each timestep
!    Rainh(i): difference in cumtotal and pcumtot, array used to
find maximum rainfall in a timestep
!    Raterain: instantaneous rainfall/def == rainfall intensity
at timestep **Calculated using DEF as the denominator
!    Smalle: rainfall energy term
!    BigE: energy term of cumulative energy at each time step
!    Raimax: maximum rainfall in a timestep
!    Raimax30: maximum rainfall in 30 minutes
!    Rtpeak: time where maximum rainfall in a timestep occurs
!    Rainh30: 30 minute peak intensity
!    Rfi1: Rainfall intensity calculation ** Calculated using
dtime as denominator
!    Rfi(i): intensity (Rfi1) converted from mm/hr to m/s and
stored in array
!    Ref(nref,1): matrix storing actual time in column 1
!    Ref(nref,2): matrix storing excess rainfall, mm
!    Refcum: used to store flow from previous timestep to
subtract it in next time step,
!    Rfix: maximum rainfall intensity converted to m/s
!    Rfix30: maximum rainfall intensity mm per 30 minutes
converted to mm/hr
!    RI30: maximum 30 minute intensity mm/hr converted to
inch/hr

C --------------------------------------------------------------
-------------
c  Storm type I and IA - fitted equations from tabular data on
Haan's
C --------------------------------------------------------------
-------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
-------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      CHARACTER*4 stype(5)

COMMON/rain/rfix,rti(5000),rfi(5000),rcum(5000,2),ref(5000,2),nc
um
      DIMENSION rainint(5000),rainint1(5000),rti1(5000)
      DIMENSION rainh(5000),rainh30(5000)
      DATA stype/'I  ','IA ','II ','III','user'/
```

```
      pd = P
c ---lw 5/30/2022 - hyetograph using time step from green-ampt
      dtime = 5.d0/60.d0
      ndtime=INT(D/dtime+1) !number of timesteps given storm
duration
      !print*,'ndtime',ndtime
C ---------------------------------------------------------------
-------
      pcumtot=0.d0
      refcum=0.d0
      IFlag=0
      ti=0.d0
      bigE=0.d0
      raimax=0.d0
      raimax30=0.d0
      nref=0
      ref=0.d0

C ---v3 09/2011 rmc
C -- calculate scaling factors for D<24 h, based on eq. 3-7,
Haan et.al. (1994)
c**> set Cooly (1980) a1,b1 coef.

      IF (stype(jstype).eq.'I  ') THEN
         tmid=9.995d0
         a1 = 15.03d0
         b1 = 0.578d0
      ELSE IF (stype(jstype).eq.'IA ') THEN
         tmid=7.96d0
         a1 = 12.98d0
         b1 = 0.7488d0
      ELSE IF (stype(jstype).eq.'II ') THEN
         tmid=11.8d0
         a1 = 17.9d0
         b1 = 0.4134d0
      ELSE IF (stype(jstype).eq.'III ') THEN
         tmid=12.d0
         a1 = 21.51d0
         b1 = 0.2811d0
      ELSE
         tmid=rcum(1,1)
      END IF
C --  scaling factors for other storm durations and volume
      tminus=tmid-d*.50d0
      tplus=tmid+d*.5d0
      pm=SCStorm(jstype,tminus)
```

```
      pp=SCStorm(jstype,tplus)

C -- rain time step loop ---
      DO 3 i=1,ndtime
         smalle=0.d0
         rti(i)=(i-1)*dtime
         tsmall=tmid+rti(i)-d*.5d0
         ptp=SCStorm(jstype,tsmall)
          !print*,'pd,ptp,pm,pp',pd,ptp,pm,pp
C --------------------------------------------------------------
-------------------------------------------------------------
c  Calculate Cumulative Hyetograph for any duration and volume
C --------------------------------------------------------------
-------------------------------------------------------------
         cumtotal=pd*(ptp-pm)/(pp-pm)
         !print*,'cumtotal,ixa',cumtotal,xIa
         !print*,'cumtotal',cumtotal
         !print*,'xIa',xIa
c        WRITE(*,'(6f9.4)')rti(i),ptp,cumtotal,
c    C        SCStorm(jstype,rti(i)),SCStorm(3,rti(i))
c        WRITE(*,'(6f9.4)')rti(i),pd,ptp,pm
C        WRITE(NUT,*),'(6f9.4)')rti(i),ptp,cumtotal,
C        SCStorm(jstype,rti(i)),SCStorm(3,rti(i))
C        WRITE(NUT,'(6f9.4)')rti(i),pd,ptp,pm
         IF(cumtotal.gt.xIa.and.IFlag.eq.0) THEN
            IFlag=1
            ti=(rti(i)-rti(i-1))/(cumtotal-pcumtot)*
     C      (xIa-pcumtot)+rti(i-1)
         END IF
C --------------------------------------------------------------
-------------------------------------------------------------
c  Calculate instantaneous hyetograph and rainfall energy term
for USLE
C --------------------------------------------------------------
-------------------------------------------------------------
         rainh(i)=cumtotal-pcumtot
         IF (rainh(i).gt.0.d0) THEN
c ---> english units ft-tons/acre-inch
            IF ((rainh(i)/25.4d0/dtime).gt.3.d0) THEN
            smalle=1074.d0
            ELSE smalle=(rainh(i)/25.4d0)*
     C      (916.d0+331.d0*dlog10(rainh(i)/25.4d0/dtime))
            END IF
c           smalle=
c     1      1099.d0 * (1.d0-0.72*exp(-
1.27*(rainh(i)/25.4d0/dtime)))
c           PRINT*,rti(i),smalle
```

```
            bigE=bigE+smalle
c ---> metric units
c           bigE=bigE+11.9d0+8.73d0*dlog10(rainh(i)/dtime)
         END IF
         IF (rainh(i).gt.raimax) THEN
            raimax=rainh(i)
            rtpeak=rti(i)
         END IF
C --lw - I30 calculation after Chow et al, 1987, assuming
timestep of 5 minutes
         IF(i.gt.5) THEN
            rainh30(i)=rainh(i-5)+rainh(i-4)+rainh(i-3)+rainh(i-
2)
     &       + rainh(i-1)+rainh(i)
         END IF
         IF (rainh30(i).gt.raimax30) THEN
            raimax30=rainh30(i)
            rtpeak30=rti(i)
         END IF
         pcumtot=cumtotal
         rfi1=rainh(i)/dtime
         rfi(i)=rfi1/3600.d0/1000.d0 !intensity converted to m/s
and stored in array

C --rmC 08/24/11-- excess rainfall hyetograph for tabular
hydrograph
         IF(cumtotal.ge.xIa) THEN
            nref=nref+1
            ref(nref,1)=rti(i)
            ref(nref,2)=(cumtotal-
xIa)**2.d0/(cumtotal+19.d0*xIa)-refcum
            refcum=(cumtotal-xIa)**2.d0/(cumtotal+19.d0*xIa) !
modified 8.7.2023 by LW to account for different initial
abstraction calcs
         END IF
c     WRITE(*,202)rti(i)*3600,rfi(i),cumtotal,refcum,ref(nref,2)
c
WRITE(10,202)rti(i),rainh(i),tsmall,ptp,cumtotal,rfi1,smallE
C      WRITE(NUT,202)
rti(i)*3600,rfi(i),cumtotal,refcum,ref(nref,2)
C      WRITE(NUT,202)
rti(i),rainh(i),tsmall,ptp,cumtotal,rfi1,smallE
c202  FORMAT(2x,f8.2,2x,e8.3,2x,f7.3,2x,f7.3,2x,f7.3,2x,f7.3,2x,
c     C            f7.3)
3     CONTINUE
C --rmc-08/24/11-- number of hyetograph steps
      nhyet=i-1
```

```
C ---rmc 03/11/99---
      rfix=raimax/(dtime*3600.d0)/1000.d0
      rfix30=raimax30*2.d0
      rI30=rfix30/25.4d0
      !print*,'cumtotal',cumtotal
      !print*,'rfix,raimax, raimax30,rfix30,ri30',rfix,raimax,
!     & raimax30,rfix30,ri30

!
!C ---rmc 08/24/11-- DOne hyet - computing musle param
!C -------------------------------------------------------------
----------------------------------------------------------------
---------
!c  compute R for musle
!c    er=> Foster et al. 1977b, units N/h
!c    er1=> Williams, units Mg h/ha N
!C -------------------------------------------------------------
----------------------------------------------------------------
---------
!c**convert bigE to SI metric - multiply by 1.702 / 100
!c** units Rst=N/h

!c**   Cooley (1980) -> er for design storms, EI/100 = R ft
tonsf/ac
!c**                   1/0.67 * R for J/m^2
!      erCooly=a1*(P/25.4d0)**(2.119d0*rti(ndtime)**0.0086d0)
!     C           /(rti(ndtime)**b1)

C --PRINT hyetograph results 25 time steps only
      maxstep=24
      nWRITE=ndtime/maxstep !! ??? nWRITE OF MWRITE
      crainh=0.d0
      cref=0.d0
      iref=nhyet-nref

C --------------------------------------------------------------
----------------------------------------------------------------
--------
c OUTPUT - FORMAT
C --------------------------------------------------------------
----------------------------------------------------------------
--------
      !WRITE(10,5)Def*60.d0,nhyet !MAC 04/10/12
C     WRITE(NUT,5)Def*60.d0,nhyet
5     FORMAT(/,3x,'SCS ',f4.1,'-MIN HYETOGRAPH (25 of',i5,
     C  1x,'steps PRINTed)',/,/,
```

```
     C
2x,'No.',3x,'Time(hr)',3x,'Rainfall(mm)',1x,'Rain30(mm)',
     C  1x,'Eff.rain(mm)')

10     FORMAT(2x,i3,2x,f8.3,3x,3f10.3)

15     FORMAT(/,2x,'Computed Total Rain =',f10.1,' mm'/,
     C   2x,'  Actual Total Rain =',f10.1,' mm',/,
     C   2x,'  Total Rain Excess =',f10.1,' mm',/,
     C   2x,'  raimax30          =',f10.1,' mm',/,
     C   2x,'  I30               =',f10.1,' mm/h')

20     FORMAT(f6.3,x,f6.3,x,f6.3)

       RETURN
       END SUBROUTINE hyetgh


c ----------------------------------------------------------------
---------------------
c      function scstorm(jstype,ptime)
c ----------------------------------------------------------------
---------------------
c ! scs design storm type equation using generalized
coefficients
c ! (munoz-carpena and parsons,2004) for 24 hour storms,
c !     p(t)       t-b (    d    ) ^g
c !     ---- = a + ----(---------)
c !     p24         c  (e|t-b|+f )
c ----------------------------------------------------------------
---------------------
c   declare variables
c ----------------------------------------------------------------
---------------------
!      implicit double precision (a-h,o-z)
!
common/rain/rfix,rti(5000),rfi(5000),rcum(5000,2),ref(5000,2),nc
um
!      dimension cff(4,7)
!c ---------------------------------------------------------------
----------------------
!      data
cff/0.4511d0,0.3919d0,0.495d0,0.5d0,9.995d0,7.96d0,11.8d0,
!     c   12.d0,1d0,1.d0,0.56d0,24.d0,-
0.1617d0,0.843d0,10.6d0,24.04d0,
!     c   -
3.0163d0,120.39d0,130.d0,2.d0,0.013d0,0.3567d0,0.525d0,
```

```
!     c   0.04d0,0.5853d0,0.4228d0,0.75d0,0.75d0/
!
!c      do 10 i=1,4
!c         write(*,100)(cff(i,j), j=1,7)
!c         write(nut,100)(cff(i,j), j=1,7)
!c10    continue
!
!       if(jstype.le.4) then
!          cffa=cff(jstype,1)
!          cffb=cff(jstype,2)
!          cffc=cff(jstype,3)
!          cffd=cff(jstype,4)
!          cffe=cff(jstype,5)
!          cfff=cff(jstype,6)
!          cffg=cff(jstype,7)
!          bigt= ptime-cffb
!          denom=cffe*dabs(bigt)+cfff
!          if(jstype.eq.1.and.denom.ge.0.d0) then
!             scstorm=0.5129d0
!          else
!             scstorm=cffa+(bigt/cffc)*(cffd/denom)**cffg
!          end if
!       else
!          do 15 i=2,ncum
!             t1=rcum(i,1)
!             rcum1=rcum(i,2)
!             t2=rcum(i+1,1)
!             rcum2=rcum(i+1,2)
!             if(ptime.gt.t1.and.ptime.le.t2) then
!             scstorm=(ptime-t1)/(t2-t1)*(rcum2-
!     c       rcum1)+rcum1
!             end if
!15        continue
!       end if
!
!c100   format(7f9.4)
!
!        return
!        end function scstorm
c ----------------------------------------------------------------
---------------------
```

```
C      PROGRAM 18  ! Based on Hromadka book pag 222
C --------------------------------------------------------------
---------
      SUBROUTINE MOVE(DAYQI,DAYQO,DAYDS)
C --------------------------------------------------------------
---------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCC
C THIS SUBROUTINE MOVES STREAM NA FORWARD IN TIME BY DELT HOURS
C
C VARIABLES:
C
C NA:     Stream "A" number. This stream is the one to be
modeled         C
C DELT:   Duration of the translation,hrs [0.1 - 48]
C
C TIME1:  Time for Beginning of results (hrs)
C
C TIME2:  Time for End of results (hrs)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCC
C --------------------------------------------------------------
---------
C  DECLARE VARIABLES
C --------------------------------------------------------------
---------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/INPUTS/DATAINP(100,100)
      COMMON/BLK10/B(5555)
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)
      DIMENSION
A(5555),DAYQI(5555,100),DAYQO(5555,100),DAYDS(5555,100),
     & SUMQI24(5555),SUMQO24(5555)
C --------------------------------------------------------------
----------
C   READ INPUTS
C --------------------------------------------------------------
----------
      NA=INT(DATAINP(JCOUNT,4))
      DELT=DATAINP(JCOUNT,5)
      TIME1=DATAINP(JCOUNT,6)
      TIME2=DATAINP(JCOUNT,7)
```

```
      AREA=DATAINP(JCOUNT,8)
C --INITIALIZE SUM STORAGE ARRAYS
      SUMQI24=0.D0
      SUMQO24=0.D0
C ---------------------------------------------------------------
----------
C   BEGIN PROCESS
C ---------------------------------------------------------------
----------
      CALL MREAD(NA,A)
C --- Calculate and write inflow volume (in mm) to storage
matrix
      SUMQI24(1)=0.5D0*A(1)*5.D0*60.D0*0.0283168D0
      DO 10 I=2,288
          SUMQI24(I)=SUMQI24(I-1)+0.5d0*(A(I-1)+
     &               A(I))*5.d0*60.d0*0.0283168d0 !Calculating
the sum by taking average of timesteps, multiplying over time
step to get volume, and converting to cubic meters
10    CONTINUE
      DO 20 J=1,100
            IF (INT(DPRECIP(1,J)).EQ.NZ2) THEN
               INODE=J
            END IF
20    CONTINUE
      DAYQI(JDAY,INODE)=SUMQI24(288)  ! Daily inflow volume is
equal to the sum at timestep 288
C-----------------------------------------------------------------
------
      WRITE(NUT,901)NA,DELT
      WRITE(NUT,903)NA,NA,DELT
      DO 30 I=1,5555
         B(I)=0.d0
30    CONTINUE
      NUMBER=INT(A(5555))
      !IF (NUMBER.GT.0.D0) THEN ! added by lw 10.26.2022
          XM=DELT*12.d0
          M=INT(XM)
          TIME=0.d0
C M=NUMBER OF INTERVALS MOVED FORWARD
          NUM1=NUMBER+M
          IF(NUM1.GE.5555)THEN               !! REMOVED by lw
11.14.2022 while testing how number is affecting flow of tails
!C HYDROGRAPH EXCEEDS 576; REDUCE NUMBER
!              NUMBER=NUMBER+(576-NUM1)
             NUM1=5555-1
!C MOVE HYDROGRAPH FORWARD
          ENDIF
```

```
          XA=M
          XA=XM-XA
          XB=1.d0-XA
          DO 100 I=1,NUMBER
              J=I+M+1
              JJ=I+M
              B(J)=A(I)*XA
              B(JJ)=A(I)*XB+B(JJ)
              TIME=TIME+.083333d0
              IF (I.GT.1) THEN
                  SUMQO24(I)=SUMQO24(I-1)+0.5d0*(B(I-1)+
     &                B(I))*5.d0*60.d0*0.0283168d0
              END IF

IF(TIME.GE.TIME1.AND.TIME.LE.TIME2)WRITE(NUT,921)TIME,
     c            A(I),B(I)
100       CONTINUE
          IF (NUMBER.GE.288.D0) THEN
              DAYQO(JDAY,INODE)=SUMQO24(288)
          ELSE
              DAYQO(JDAY,INODE)=SUMQO24(288)
          END IF
          DAYDS(JDAY,INODE)=DAYQI(JDAY,INODE)-DAYQO(JDAY,INODE)
C ----------------------------------------------------------------
----------
C WRITE RESULTS TO MEMORY
C ----------------------------------------------------------------
----------
          B(5555)=J
          CALL MWRITE(NA,B)
      !ELSE
      !    WRITE(nut,999)
      !END IF
C ----------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C ----------------------------------------------------------------
----------
901   FORMAT(/,10X,'MOVE STREAM NUMBER',I2,' FORWARD IN TIME',
     C  ' BY',F7.3,' HOURS:',/)

903   FORMAT(10X,' MODEL        STREAM',I2,3X,' STREAM',I2,/,
     C  10X,' TIME          (CFS)      MOVED',F7.3,' HOURS')
999   FORMAT(10X,'NO FLOW OCCURRED ON THIS DAY')
921   FORMAT(10X,F7.3,3X,F10.1,F10.1)
C ----------------------------------------------------------------
----------
```

```
C END PROCESS
C --------------------------------------------------------------
----------
      RETURN
      END SUBROUTINE MOVE
```

```
C ---------------------------------------------------------------
--------
C  PROGRAM 21 - Based on Hromadka book
C---------------------------------------------------------------
-----------------
c    Date      Modification
Initials
c   -------    ------------------------------                ---
---
c   5/29/20    Read values from SNODE
llw
C ---------------------------------------------------------------
--------
      SUBROUTINE MREAD(icol,TEMP)
! Fix TEMP values to icol column of SS
!---------------------------------------------------------------
--------
! INTERNAL VARIABLES
! TEMP: STORAGE MATRIX TO PASS TO FUNCTION
! ICOL: INDICATES COLUMN TO READ FROM SS
! SS: STORAGE MATRIX FOR FLOWS AT EACH STREAM
! --------------------------------------------------------------
--------
C ---------------------------------------------------------------
--------
C   DECLARE VARIABLES
C ---------------------------------------------------------------
--------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)
      DIMENSION TEMP(5555)
C ---------------------------------------------------------------
--------
      DO 100 I=1,5555
         TEMP(i)=SS(i,icol)
100   CONTINUE

      RETURN
      END SUBROUTINE MREAD
```

```
C --------------------------------------------------------------
----------
C  PROGRAM MUSLE
C --------------------------------------------------------------
----------
      SUBROUTINE
musle(er,er1,erCoolm,ek,Y,pl,cfact,pfact,Area,Q,tc,P,
     C
D,isoil,dp,sconc,sconc1,sconc2,om,aa1,b1,bigE,raimax30,qp,
     C      ieroty,sconc3)
C --------------------------------------------------------------
----------
C   version 3.0.1, Last ModIFied: See ModIFications below
C   WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C   Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons, BAE
(jep)
C               University of Florida        BAE, NC State
University
C               Gainesville, FL 32611        Raleigh, NC 27695-
7625(USA)
C              e-mail: carpena@ufl.edu
C --------------------------------------------------------------
---------------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
---------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      CHARACTER*20 typesoil(21)
      DIMENSION
types(21),d50(21),sand(21),silt(21),Tf(21),Sf(21),Pf(21)
C --------------------------------------------------------------
----------
      DATA typesoil/'Clay','Silty clay','Sandy clay','Silty clay
loam',
     C  'Clay loam','Sandy clay loam','Silt','Silt loam','Loam',
     C  'Very fine sandy loam','Fine sandy loam','Sandy loam',
     C  'Coarse sandy loam','Loamy very fine sand','Loamy fine
sand',
     C  'Loamy sand','Loamy coarse sand','Very fine sand',
     C  'Fine sand','Sand','Coarse sand'/
      DATA types/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
     C         18,19,20,21/
     DATA d50/23.d0,24.d0,66.d0,25.d0,
     C         18.d0,91.d0,19.d0,27.d0,35.d0,
```

```
     C         35.d0, 80.d0,98.d0,
     C         160.d0,90.d0,120.d0,
     C         135.d0,180.d0,140.d0,160.d0,170.d0,200.d0/
     DATA sand/20.d0,10.d0,50.d0,15.d0,35.d0,
     C          55.d0,5.d0,20.d0,45.d0,60.d0,
     C          60.d0,60.d0,60.d0,84.d0,84.d0,
     C          84.d0,84.d0,90.d0,90.d0,90.d0,
     C          90.d0/
     DATA silt/30.d0,45.d0,10.d0,50.d0,30.d0,
     C          20.d0,85.d0,60.d0,35.d0,25.d0,
     C          25.d0,25.d0,25.d0,8.d0,8.d0,
     C           8.d0,8.d0,5.d0,5.d0,5.d0,
     C           5.d0/
     DATA tf/0.01287d0,0.01870d0,0.01714d0,0.02606d0,0.0236d0,
     C        0.02778d0,0.05845d0,0.04259d0,0.03618d0,0.03877d0,
     C        0.03205d0,0.02549d0,0.01914d0,0.03726d0,0.02301d0,
     C        0.01624d0,0.00982d0,0.04401d0,0.02173d0,0.01481d0,
     C        0.00827d0/
     DATA sf/0.065d0,0.065d0,0.065d0,0.065d0,0.065d0,
     C        0.065d0,0.065d0,0.065d0,0.0325d0,-0.035d0,
     C        0.d0,0.0325d0,0.0325d0,-0.0325d0,0.d0,
     C        0.0325d0,0.0325d0,-0.0325d0,0.d0,0.0325d0,
     C        0.0325d0/
     DATA pf/0.075d0,0.075d0,0.075d0,0.050d0,0.050d0,
     C        0.05d0,0.025d0,0.025d0,0.025d0,0.d0,
     C        0.d0,0.d0,0.d0,-0.025d0,-0.025d0,
     C        -0.025d0,-0.025d0,-0.05d0,-0.05d0,-0.05d0,
     C        -0.05d0/
C -------------------------------------------------------------
----------
C  Compute R for musle
C -------------------------------------------------------------
----------
c     er=> Foster et al. 1977b, units
c     er1=> Williams, units Mg h/ha N
C -------------------------------------------------------------
----------
c**convert bigE to SI metric - multiply by 1.702/100
c** units Rst=N/h, Runoff volume: vol(m3), volro (mm)
C -------------------------------------------------------------
----------

          volro = Q
          vol=volro*(Area*10000.d0/1000.d0)
          !print*,'Q in musle',Q
          !print*,'volro',vol
          !print*,'vol',vol
```

```
          qpdepth=qp*360.d0/Area
          Def=0.24d0*tc
          dtime=Def
          ndtime=INT(D/dtime+1.d0)
          bigEm=0.006700d0*bigE

rst=1.702d0*(bigE/100.d0)*(raimax30/25.4d0/dtime)*dtime/0.5d0
          rro=volro*(qpdepth)**(1.d0/3.d0)
          er=0.5d0* rst + 0.35d0*rro
          er1=9.05d0*(vol*qp)**0.56d0/Area
          rain = P
C rmc03/28/99-- er1 = 9.05d0*(vol*qp)**0.56d0
C -------------------------------------------------------------
----------
c**   Cooley (1980) -> er for design storms, EI/100 = R ft
tonsf/ac
c**                   1/0.67 * R for J/m^2
C -------------------------------------------------------------
----------

erCooly=aa1*(rain/25.4d0)**(2.119d0*D**0.0086d0)/(D**b1)
          erCoolm=erCooly*1.702d0
C -------------------------------------------------------------
----------
c  erGLEAMS, from GLEAMS daily rain
C -------------------------------------------------------------
-------------
          gei=7.87d0*(rain/25.4d0)**1.51d0
          geim=1.702d0*gei

C -------------------------------------------------------------
-------------
      WRITE(NUT,17)
C -------------------------------------------------------------
----------
      IF (ieroty.eq.1)THEN !!       1) Williams (1975)
         WRITE(NUT,33) Area,vol,qp,er1
      ELSE IF (ieroty.eq.2)THEN !! 2) GLEAMS / daily CREAMS
         WRITE(NUT,32) rain,gei,geim
      ELSE IF (ieroty.eq.3)THEN !! 3) Foster et al. (1977)
         WRITE(NUT,22) bigE,bigEm,volro,qpdepth,rst,rro,er
      ELSE IF (ieroty.eq.4)THEN !! 4) Cooley (1980) - Design
Storm
         WRITE(NUT,24) aa1,b1,rain,D,erCooly, erCoolm
      END IF
C -------------------------------------------------------------
-------
```

```
c K-FACTOR (calculate internally IF user did set -1 in input
file)
C -----------------------------------------------------------------
-------
      IF(dp.le.0.d0) THEN
         dp=d50(isoil)
      END IF
c    om=2.d0
      IF ((ek.lt.0.d0)) THEN
         ek=tf(isoil)*(12.d0-om)+sf(isoil)+pf(isoil)
c --  convert to metric units - kg/N * h/m^2
          ek=0.1317d0*ek
      END IF
c***  save english version
      ekeng=ek/0.1317d0
c---  Write table of results
!     WRITE(NUT,98)
!     DO i=1,21
!
WRITE(10,99)i,types(i),sand(i),silt(i),tf(i),sf(i),pf(i),d50(i)
!
WRITE(NUT,99)i,types(i),sand(i),silt(i),tf(i),sf(i),pf(i),d50(i)
      WRITE(NUT,99)typesoil(isoil),sand(isoil),silt(isoil),
     C    tf(isoil),sf(isoil),pf(isoil),d50(isoil),om,ek,ekeng
!     END DO
C -----------------------------------------------------------------
-------
c S-FACTOR
C -----------------------------------------------------------------
-------
      theta=DATAn(Y)
      s=dsin(theta)
c ** Usle
c      bigS=65.4d0*s**2+4.56d0*s+0.065d0
c from haan p261
      IF (s.lt.0.09d0) THEN
         bigS=10.8d0*s +0.03d0
      ELSE
         bigS=16.8d0*s -0.5d0
      END IF
      IF (pl.lt.0.7d0) THEN
         bigS=3.0d0*s**0.8d0+0.56d0
      END IF
C -----------------------------------------------------------------
-------
c L-FACTOR after McCool, p262 Haan
```

```
C --------------------------------------------------------------
-------
c      IF (x.lt.3.d0) THEN
c         x=0.3d0
c      ELSE IF (x.eq.4.d0) THEN
c         x=0.4d0
c      ELSE
c         x=0.5d0
c      END IF
c * use distance not length along slope
      slopeL = pl*cos(theta)
      beta=11.16d0*s/(3.d0*s**0.8d0+0.56d0)
      x=beta/(1.d0+beta)
      bigL=(slopeL/22.d0)**x
C --------------------------------------------------------------
-------
c Final sediment yield calculations
C --------------------------------------------------------------
-------
      A0=er*ek*bigL*bigS*cfact*pfact
      A1=er1*ek*bigL*bigS*cfact*pfact
      A2=geim*ek*bigL*bigS*cfact*pfact
      A3=erCoolm*ek*bigL*bigS*cfact*pfact
C --------------------------------------------------------------
-------
C Concentrations of Sediment in g/L
C --------------------------------------------------------------
-------
      IF(volro.le.0)THEN
         sconc=0.d0
         sconc1=sconc
         sconc2=sconc
         sconc3=sconc
      ELSE
         sconc= A0*Area*10000.d0/vol
         sconc1=A1*Area*10000.d0/vol
         sconc2=A2*Area*10000.d0/vol
         sconc3=A3*Area*10000.d0/vol
      END IF
C --------------------------------------------------------------
-------
c  PRINT SUMMARY
C --------------------------------------------------------------
-------
c      WRITE(NUT,19) isoil,ek,ekeng,om,dp
c      WRITE(NUT,21) slopeL,beta,bigL,bigS,cfact,pfact
c      WRITE(NUT,21) slopeL,beta
```

```
c      WRITE(NUT,23)
C --------------------------------------------------------------
-------
C   CONVERT kg/m^2 to Eng Units ton/ac
C --------------------------------------------------------------
-------
      ccc=0.00110231131d0/0.000247105381d0
C --------------------------------------------------------------
-------
   !! IF (ieroty.eq.1)THEN !!     1) Williams (1975)
   !!     WRITE(NUT,26)'Rw
(Williams)',A1,A1*ccc,sconc1,er1,ek,bigL,
   !!C       bigS,cfact,pfact
   !!   ELSEIF (ieroty.eq.2)THEN !! 2) GLEAMS / daily CREAMS
   !!     WRITE(NUT,26)'Rm (GLEAMS)
',A2,A2*ccc,sconc2,geim,ek,bigL,
   !!C       bigS,cfact,pfact
   !!   ELSEIF (ieroty.eq.3)THEN !! 3) Foster et al. (1977)
   !!    WRITE(NUT,26)'Rm (Foster)
',A0,A0*ccc,sconc,er,ek,bigL,
   !!C       bigS,cfact,pfact
   !!   ELSEIF (ieroty.eq.4)THEN !! 4) Cooley (1980) - Design
Storm
   !!    WRITE(NUT,26)'Rst (Cooley)
',A3,A3*ccc,sconc3,ercoolm,ek,
   !!C       bigL,bigS,cfact,pfact
   !! ENDIF
C --------------------------------------------------------------
-------
      IF (ieroty.eq.1)THEN !!     1) Williams (1975)
         WRITE(NUT,26)bigL,bigS,cfact,pfact,A1,A1*ccc,sconc1
      ELSEIF (ieroty.eq.2)THEN !! 2) GLEAMS / daily CREAMS
         WRITE(NUT,26)bigL,bigS,cfact,pfact,A2,A2*ccc,sconc2
      ELSEIF (ieroty.eq.3)THEN !! 3) Foster et al. (1977)
         WRITE(NUT,26)bigL,bigS,cfact,pfact,A0,A0*ccc,sconc
      ELSEIF (ieroty.eq.4)THEN !! 4) Cooley (1980) - Design
Storm
         WRITE(NUT,26)bigL,bigS,cfact,pfact,A3,A3*ccc,sconc3
      ENDIF

C --------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C --------------------------------------------------------------
----------
17    FORMAT(/,1x,76('-'),/,4x,'MUSLE SOIL EROSION
CALCULATIONS',/,
```

```
     C  1x,76('-'))
33    FORMAT(/,2x,'Rw Williams (1975)',
     C  /,4x,'Watershed area     =  ',f15.3,' ha',
     C  /,4x,'Volume of runoff   =  ',f15.2,'  m3',
     C  /,4X,'Qpeak              =  ',f15.4,' m3/s',
     C  /,4x,'Rw (Williams)      =  ',f15.4,' N/hr')
32    FORMAT(/,2x,'Rw GLEAMS / daily CREAMS',
     C   /,4x,'Rain  =',f10.2,' mm',
     C   /,4x,'R_GLM =',f10.2,' From GLEAMS - Wischmeier',
     C   /,4x,'R_GLM =  ',f10.4,' N/h - Converted to Metric')
22    FORMAT(/,2x,'Rw Foster et al. (1977)',
     C   /,4x,'E = ',f10.3,' ft-tonf/acre =',f10.3,' MJ/ha',
     C   /,4x,'Volume Runoff =',f10.4,' mm',
     C   /,4X,'Qpeak         =',f10.4,' mm/hr',
     C   /,4x,'Factors in Rm:'
     C   /,7x,'Rstorm      =',f10.4,
     C   /,7x,'Rrunoff     =',f10.4,
     C   /,7x,'Rm (Foster) =',f10.4,' N/hr')
24    FORMAT(/,2x,'Rw Cooley (1980) - Design Storm ',
     C   /,4x,'a1   = ', f10.4,
     C   /,4X,'b1   = ',f10.4,
     C   /,4x,'Rain =',f10.3, ' mm ',
     C   /,4X,'D    =',f10.3,' hr',
     C   /,4x,'Rst  =',f10.3,' ft-tonf/acre =',f10.3,' N/ha')
99    FORMAT(/,4x,'Soil type          =  ',A20,
     C  /,4x,'Sand and silt      =  ',2f10.4,' %',
     C  /,4x,'Particle size (d50)=  ',f10.4,' um',
     C  /,4x,'Oganic matter (OM) =  ',f10.4,' %',
     C  /,4x,'USLE text. factor  =  ',f10.4,
     C  /,4x,'USLE struc. factor =  ',f10.4,
     C  /,4x,'USLE perc. factor  =  ',f10.4,
     C  /,4x,'USLE K factor      =  ',f10.4,' kg-h/N-m^2',f10.4,
     C' Eng.')

26    FORMAT(4x,'USLE L factor      =  ',f10.4,
     C  /,4x,'USLE S factor      =  ',f10.4,
     C  /,4x,'USLE C factor      =  ',f10.4,
     C  /,4x,'USLE P factor      =  ',f10.4,
     C  //,4x,'Soil loss A        =  ',f10.4,'
kg/m^2',4x,f10.4,' t/ac',
     C  /,4X,'Mean day sed. conc.=  ',f10.4,' g/l',/)
23    FORMAT(/,4X,100('-'),/,6x,'Method',14x,'Soil Loss A',
     C  13x,'Sediment ',4x,'L-Factor',2x,'S-Factor',2x,'C-
Factor',
     C  2x,'P-Factor',/,2x,18x,' kg/m^2',4x,'EngUnits t/ac',
     C  6x,'Conc g/l',/,4x,100('-'))
25    FORMAT(
```

```
     C  2x,A13,2x,f10.2,4x,f10.2,4x,f10.2,2x,f10.2)
c26    FORMAT(
c     C  4x,A13,2x,f10.2,4x,f10.2,4x,f10.2,2x,f10.2,3x,f7.3,
c     C  4(2x,f8.3))
98    FORMAT(/,4x,'Table for Computing Ksoil (from GLEAMS and
KINEROS)',
     C /,4X,100('-'),/,4x,' i',4x,'Soil
Type',10x,'Sand',3x,'Silt',3x,
     C 'Tex.F.',4x,'Str.F.',4x,'Per.F.',4x,'D50',5x,'OM',6x,
     C 'K',8x,'K(Eng)',/,30x,'[%]',4x,'[%]',33x,'[um]',4x,'[%]',
     C 2x,'kg-h/N-m^2',/,4X,100('-'))
c99
FORMAT(4x,i2,4x,a20,f4.0,2x,f4.0,2x,f8.5,2x,f8.4,2x,f7.3,2x,
c     C  f6.1,2x,f7.3,2x,f7.3,2x,f7.3)

      RETURN
     END SUBROUTINE musle
```

```
C ----------------------------------------------------------------
---------------
C PROGRAM 21 - Based on Hromadka book
C-----------------------------------------------------------------
----------------
c    Date      Modification
Initials
c   -------    ------------------------------              ---
---
c   2/17/99    Check for 0.1<Ia/P<0.5
rmc
C   5/29/20    Write to SNODE
llw
C ----------------------------------------------------------------
---------------
      SUBROUTINE MWRITE(icol,TEMP)
C For each column (icol) of holding matrix (SS), fixes value to
TEMP vector
!-----------------------------------------------------------------
--------
! INTERNAL VARIABLES
! TEMP: STORAGE MATRIX TO PASS TO FUNCTION
! INODE: INDICATES NODE/COLUMN TO READ FROM SNODE
! DPRECIP: READS WHICH NODE SHOULD BE USED FOR THIS PROCESS
! SNODE: STORAGE MATRIX FOR FLOWS AT EACH NODE
! JCOUNT: COUNTER FOR READING INPUT FILE ROW
! SS: ST
C ----------------------------------------------------------------
----------------------------
C   DECLARE VARIABLES
C ----------------------------------------------------------------
----------------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)
      DIMENSION TEMP(5555)
C ----------------------------------------------------------------
---------------
C      WRITE(*,*)(DPRECIP(1,J),J=1,100)
      DO 50 K=1,100
        IF (INT(DPRECIP(1,K)).EQ.NZ2) THEN
            inode=K
        END IF
50    CONTINUE
```

```
      DO 100 I=1,5555
         SNODE(i,inode)=TEMP(i)
         !IF (KODE.LT.11) THEN
          SS(i,icol)=TEMP(i)
         !END IF
100   CONTINUE

      RETURN
      END SUBROUTINE MWRITE
```

```
! --------------------------------------------------------------
--------
!  BASED ON PROGRAM 21 MREAD - Based on Hromadka book
!---------------------------------------------------------------
-----------------
!    Date      Modification
Initials
!   -------    ------------------------------              ---
---
!   4/28/22    Read values from SNODE
llw
! --------------------------------------------------------------
--------
      SUBROUTINE NREAD(NZ,TEMP)
! Fix TEMP values to icol column of SS
!---------------------------------------------------------------
--------
! INTERNAL VARIABLES
! TEMP: STORAGE MATRIX TO PASS TO FUNCTION
! INODE: INDICATES NODE/COLUMN TO READ FROM SNODE
! --------------------------------------------------------------
--------
!   DECLARE VARIABLES
! --------------------------------------------------------------
--------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      DIMENSION TEMP(5555)
C --------------------------------------------------------------
--------
      DO 50 J=1,100
         IF (INT(DPRECIP(1,J)).EQ.NZ) THEN
            inode=J
 !           print *,inode,j
         END IF
50    CONTINUE

      DO 100 I=1,5555
         TEMP(i)=SNODE(i,inode)
100   CONTINUE

      RETURN
      END SUBROUTINE NREAD
```

```
C  PROGRAM 14 - Based on Hromadka book pag 167
C -------------------------------------------------------------
---------------------------
      SUBROUTINE OASB (KTYPE,H,INTERV,XMAX,UNIT,SUM,TIME1,TIME2)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
C INTEGER LETM,BLANK,DOT,CROSS,DASH,LINE(41)
C
C To compile with gfortran is necesary to declare these
variables as                      C
C CHARACTER instead of INTEGER
C
C KTYPE: 24-hr storm unit-interval model number
C H: Array containing flow values BUT shifted for time? i.e.
H(1) is not equal to H at time 5 min
C INTERV: number of intervals in array containing flow values
C XMAX: Peak flow (cfs) of hydrograph
C UNIT: Unit interval of each time step (minutes)
C SUM: Cumulative flow  (ac-ft/day)
C TIME1: Time for Beginning of results (hrs)
C
C TIME2: Time for End of results (hrs)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCC
C -------------------------------------------------------------
---------------------------
C   DECLARE VARIABLES
C -------------------------------------------------------------
---------------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      DIMENSION H(5555)
      CHARACTER*1 LETM,BLANK,DOT,CROSS,DASH,LINE(41)
      DATA LETM,BLANK,DOT,CROSS,DASH,LINE/'V','
','.','Q','I',41*' '/
C -------------------------------------------------------------
---------------------------
c      WRITE(NUT,101)
      WRITE(NUT,130)
c      WRITE(NUT,101)
      WRITE(NUT,10)
c      WRITE(NUT,101)
      WRITE(NUT,130)
      STEP=5.d0*60.d0/43560.d0 !
      WRITE(NUT,103)
```

```
      WRITE(NUT,130)
      F=40.d0/XMAX
      FMASS=40.d0/SUM
      T0=0.0d0
      T1=XMAX/4.d0
      T2=2.d0*T1
      T3=3.d0*T1
      WRITE(NUT,105)T0,T1,T2,T3,XMAX
C ---------------------------------------------------------------
----------------------------
      WRITE(NUT,130)
      XMASS=0.d0
C ---------------------------------------------------------------
----------------------------
      GO TO(141,142,143,146,147,148),KTYPE
141   KNUM=1
      GO TO 144
142   KNUM=2
      GO TO 144
143   KNUM=3
      GO TO 144
146   KNUM=4
      GO TO 144
147   KNUM=6
      GO TO 144
148   KNUM=12
C ---------------------------------------------------------------
----------------------------
144   TIME=0.d0
      IF(KTYPE.EQ.0)THEN
         KNUM=1
         TIME=TIME1
         !print*,'time in oasb', time
      END IF
C ---------------------------------------------------------------
----------------------------
C     OUTPUT GRAPH LOOP
C ---------------------------------------------------------------
----------------------------
c      PRINT*,'INTERV',INTERV
c      PRINT*,'H(5555)',H(5555)
      DO 200 I=1, INTERV
         TEST=H(I)*F
c       print*,'H(I),F,TEST=',H(I),F,TEST
c         print*,'H(I),I,F,TEST=',H(I),I,F,TEST
c         PRINT*,'xmass',xmass
         XMASS=XMASS+H(I)*STEP
```

```
         LINE(1)=DOT
         LINE(11)=DOT
         LINE(21)=DOT
         LINE(31)=DOT
         LINE(41)=DOT
         J=INT(TEST)+1
         JMASS=INT(XMASS*FMASS+1)
         IF (JMASS.GT.41) GO TO 201
         IF (J.GT.41) GO TO 201
c       print*,'J,JMASS=',J,JMASS
         LINE(JMASS)=LETM
         LINE(J)=CROSS
         IF(KNUM.NE.1) THEN
c rmc 320   DO 350 K=1, KNUM
            DO 350 K=1, KNUM
               TIME=TIME+.083333d0
               IF(K.EQ.1) GO TO 349
                  XMASS=XMASS+H(I)*STEP
                  JMASS=INT(XMASS*FMASS+1)
                  IF (JMASS.GT.41) GO TO 201
                  IF (J.GT.41) GO TO 201
                  LINE(JMASS)=LETM
                  LINE(J)=CROSS
349               IF(TIME.GE.TIME1.AND.TIME.LE.TIME2)
WRITE(NUT,210)
     C              TIME,XMASS,H(I),LINE
                  LINE(JMASS)=BLANK
350        CONTINUE
c           GO TO 215
         ENDIF
         IF(TIME.GE.TIME1.AND.TIME.LE.TIME2) WRITE(NUT,210)TIME,
     C    XMASS,H(I),LINE
215         LINE(J)=BLANK
            LINE(JMASS)=BLANK
          TIME=TIME+.083333d0
200   CONTINUE
201   WRITE(NUT,130)
C --------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C --------------------------------------------------------------
----------
10    FORMAT(32X,' 24 - HOUR STORM',/,32X,'RUNOFF HYDROGRAPH')
101   FORMAT(1X,76('*'))
102   FORMAT(1X,76('-'))
130   FORMAT(1X,76('='))
```

```
103   FORMAT(/,19X, 'HYDROGRAPH IN FIVE-MINUTE INTERVALS
(CFS)',/)
105   FORMAT(3X,'TIME(HRS)  VOLUME(AF)  Q(CFS)
',F3.0,1X,4F10.1)
210   FORMAT(3X,F7.3,F12.4,1X,F9.2,2X,41A1)
989   FORMAT(/)
C --------------------------------------------------------------
----------
      RETURN
      END SUBROUTINE OASB
```

```
      SUBROUTINE PCALC(P,FIRR,DIRREFF,DAYQO,JNODE,DAREA,PEFF)
!C -------------------------------------------------------------
---------------
! This subroutine calculates effective rainfall by incorporating
ET, Baseflow, initial moisture content,
C     surface water abstractions, and irrigation into effective
rainfall
!C -------------------------------------------------------------
----------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)

common/CINPUT/DETO(5555,100),DBF(5555,100),DTAVG(5555,100),
     &
DTMAX(5555,100),DTMIN(5555,100),DWS2(5555,100),DSORAD(5555,100),
     &
DCKM(5555,100),DAB(5555,100),DIRR(5555,100),DSNO(5555,100)
      DIMENSION FIRR(1,100),PEFF(5555,100),DIRREFF(5555,100),
     & DAREA(100),AA(5555),A(5555),DAYQO(5555,100)



      DO 100 I=1,100
          IF (INT(DPRECIP(1,I)).EQ.NZ2) THEN
              Pinit=DPRECIP(JDAY+1,I)
              !ET=DETO(JDAY+1,I)
              !BASEF=DBF(JDAY+1,I)
              !WCINIT=DSM(JDAY+1,I)
              !ABSTR=DAB(JDAY+1,I)
              PIRR=DIRR(JDAY+1,I)
              FRACIRR=FIRR(1,I)
              INODE=I
          END IF
100   CONTINUE

      AREA=DAREA(INODE)
      !print*,'inode,area',inode,area
      DIRREFF(JDAY,INODE)=0.D0
      AMC=0.D0
      If (JDAY.EQ.1) THEN
          AMC = 25.4d0
      ELSE IF (JDAY.LT.4) THEN
          DO icount = 1, JDAY-1
```

```
                AMC = AMC + PEFF(ICOUNT,INODE)/(AREA*10.D0)+PINIT
            END DO
        ELSE
            AMC=(PEFF(JDAY-1,INODE)+PEFF(JDAY-2,INODE)+
     &      PEFF(JDAY-3,INODE))/(AREA*10.D0)+PINIT
        END IF
        IF (AMC.LT.25.4D0) THEN
            FRACIRR=FRACIRR
        ELSE
            FRACIRR=0.D0
        END IF

        DIRRM3=FRACIRR*PIRR*AREA*10

        IF (DAYQO(JDAY,JNODE).GT.0.D0) THEN
            !IF (DIRRM3.GT.0.75D0*DAYQO(JDAY,JNODE)) THEN
            !DIRRM3=0.75D0*DAYQO(JDAY,JNODE)
            IF (DIRRM3.GT.DAYQO(JDAY,JNODE)) THEN
            DIRRM3=DAYQO(JDAY,JNODE)
            ELSE
               DIRRM3=FRACIRR*PIRR*AREA*10
            END IF
        ELSE
            DIRRM3=0.D0
        END IF

        !print*,'pinit,dirrm3',pinit,dirrm3

        P=Pinit+DIRRM3/(AREA*10)
        IF (AREA.EQ.0.D0) THEN
            P=0.D0
        END IF
        IF (P.LE.0.D0) THEN
            P=0.D0
        END IF


c--- store irrigation amount as cubic meters for water balance
        DIRREFF(JDAY,INODE)=DIRRM3
        !print*,'irrigation',dirreff(jday,inode)
!        DIRRCFS=DIRRM3/(3600.d0*24.d0) !convert daily m3
irrigation withdrawal to cfs
!        IF (DIRRM3.GT.0.D0) THEN
!            EXCESS=0.D0
!            NUMBER=0
!            DO 10 I=1,5555-1
!            IF (I.le.288) then
```

```
!               DIRRCFS=EXCESS+DIRRCFS
!                A(I) = AA(I) - DIRRCFS
!                IF (A(I).LT.0.D0) THEN
!                    A(I)=0.D0
!                    EXCESS=DIRRCFS-AA(I)
!                END IF
!           end if
!          IF (A(I).GT.0.D0) THEN
!              NUMBER = NUMBER + 1
!          ELSE
!              NUMBER = NUMBER
!          END IF
!10         CONTINUE
!
!           A(5555)=NUMBER

!           CALL MWRITE(NA,A,NZ1)

      RETURN
      END SUBROUTINE PCALC
```

```
C   PROGRAM 20 - Based on Hromadka book 237 pag.
C -----------------------------------------------------------------
-------------
      SUBROUTINE piper(DAYQI,DAYQO,DAYDS,PSTORE) !ARGU = nut
(9.25.17)
C -----------------------------------------------------------------
-------------
C  THIS SUBROUTINE PERFORMS NORMAL DEPTH ROUTING
C -----------------------------------------------------------------
-------------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCC
C NA: Stream "A" number. This stream is the one to be modeled
C
C XL: Piper length - the length of the  longest watercourse
(FEET)          C
C XN: Basin Factor (Manning's Friction Factor) [0.008 - 0.999]
C
C E1: Upstream elevation (m) [-30 to 3000]
C
C E2: Downstream elevation (m) [-60 to 3000]
C
C D:  Piper diameter (m) [0.3-30]
C
C TIME1:  Time for Beginning of results (hrs)
C
C TIME2:  Time for End of results (hrs)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCC
C   INTERNAL VARIABLES
C
C B(): OUTFLOW FROM PIPE
C A(): INFLOW TO PIPE
C TIME: Time at calculation point
C STORE: Flow stored behind pipe (AF)
C QCAP: Maximum flow capacity of pipe (
C V: Velocity of flow in pipe
C NUMBER: Number of time steps in process
C VCAP: Maximum velocity of flow possible in pipe
C XA:
C XB: conversion factor to convert flow to AC-FT
C

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCC
```

```
C ------------------------------------------------------------------
-------------
C   DECLARE VARIABLES
C ------------------------------------------------------------------
-------------
      IMPLICIT DOUBLE PRECISION(a-h,o-z) !(8.28.18)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/INPUTS/DATAINP(100,100)
      COMMON/BLK10/B(5555)
      DIMENSION F(21,2), SUMQI24(5555), SUMQO24(5555)
      DIMENSION
A(5555),DAYQI(5555,100),DAYQO(5555,100),DAYDS(5555,100),
     & PSTORE(100)
      DATA
F/0.,.05,.1,.15,.2,.25,.3,.35,.4,.45,.5,.55,.6,.65,.7,.75,
     C  .8,.85,.9,.95,1.,0.,.52,.63,.715,.78,.832,.88,.912,
     C  .945,.97,1.,1.025,1.045,1.06,1.08,1.095,1.11,1.12,1.13,
     C  1.136,1.14/
!     EXPORT Hydrograph, nute (hours) StreamA(CFS)
C ------------------------------------------------------------------
-------------
!      SS=SS1
C ------------------------------------------------------------------
-------------
      !READ(nut,*)NA,XL,XN,E1,E2,D,TIME1,TIME2
      NA=INT(DATAINP(JCOUNT,4))
      XL=DATAINP(JCOUNT,5)
      XN=DATAINP(JCOUNT,6)
      E1=DATAINP(JCOUNT,7)
      E2=DATAINP(JCOUNT,8)
      D=DATAINP(JCOUNT,9)
      TIME1=DATAINP(JCOUNT,10)
      TIME2=DATAINP(JCOUNT,11)
      Area=DATAINP(JCOUNT,12)

C ------------------------------------------------------------------
-------------
C   CONVERSION - To convert in m^3/S
C ------------------------------------------------------------------
-------------
!      XL=XL/0.3048d0 !To convert in m^3/S
!      E1=E1/0.3048d0
!      E2=E2/0.3048d0
```

```
!      D=D/0.3048d0
C ----------------------------------------------------------------
-------------
      WRITE(NUT,901)NA
      WRITE(NUT,306)XL,XN,E1,E2,D
c rmc 305     WRITE(NUT,306)XL,XN,E1,E2,D
C ----------------------------------------------------------------
----------------
C INITIALIZE VARIABLES
C ----------------------------------------------------------------
----------------
      DO 10 I=1,5555
         B(I)=0.d0
         A(I)=0.d0
         SUMQI24(I)=0.D0
         sumqo24(I)=0.D0
10    CONTINUE
      CALL MREAD(NA,A)
      TIME=0.d0
      STORE=0.d0
C --- Calculate and write inflow volume (in mm) to storage
matrix
      DO 15 I=2,288
          SUMQI24(I)=SUMQI24(I-1)+0.5d0*(A(I-1)+
     &                A(I))*5.d0*60.d0*0.0283168d0 !Calculating
the sum by taking average of timesteps, multiplying over time
step to get volume, and converting to cubic meters
15    CONTINUE
      DO 20 J=1,100
            IF (INT(DPRECIP(1,J)).EQ.NZ2) THEN
               INODE=J
            END IF
20    CONTINUE
      DAYQI(JDAY,INODE)=SUMQI24(288)  ! normalize 24 hr flow to
mm by dividing by area and converting units

c--- Calculate physical characteristics
      S=SQRT((E1-E2)/XL)  ! Slope of pipe
      QCAP=35.628d0/XN*.013d0*S*D**2.55557d0  !Max flow capacity
of pipe
      NUMBER=INT(A(5555)) !row 5555 of A matrix is reserved for
number of timesteps in streamflow hydrograph
      !IF (NUMBER.GT.0.D0) THEN
       VCAP=QCAP/.7854d0/D/D !max velocity of pipe
       XA=XL/300.d0 !
       XB=300.d0/43560.d0
```

```
       WRITE(NUT,905)
C -------------------------------------------------------------
-----------------
C UPDATE STORE VALUE
C--------------------------------------------------------------
-----------------
       STORE=STORE+PSTORE(inode)
C -------------------------------------------------------------
-----------------
C MAIN LOOP
C -------------------------------------------------------------
-----------------

       DO 550 I=1,5555-1
           IF(I.GT.NUMBER.AND.STORE.LT..001d0)GO TO 1000 ! when
number of time steps ends
           Q=A(I)
           TIME=TIME+.083333d0
       !MAC 4/9/12 Next conditional is to consider IF before the
hydrograph
       !there are records with value=0
           IF (Q.EQ.0.d0.AND.I.LT.NUMBER) THEN !if flow is 0 but
number of timesteps hasn't been meet yet
               B(I)=0.d0
               GO TO 550
           END IF
           IF(Q.LT.QCAP.AND.STORE.LE.0.d0)GO TO 510 ! open pipe
flow condition
C -------------------------------------------------------------
-----
C PIPE IS UNDER PRESSURE
C -------------------------------------------------------------
-----
           V=VCAP !velocity equals maximum possible velocity
based on pipe size
           STORE=STORE+(Q-QCAP)*XB !storage equals previous
storage plus difference between Q and Qmax times conversion
factor
           Q=QCAP
           IF(STORE.GE.0.d0)GO TO 520 !account for extra
timesteps required to route stored flow
           Q=QCAP+STORE*145.2d0 !calculates new Q taking storage
into account
           STORE=0.d0
C -------------------------------------------------------------
-----
C OPEN FLOW
```

```
C --------------------------------------------------------------
-----
510      QQ=Q/QCAP !ratio of flow to maximum flow
           INDEX=INT(QQ*20.d0+1.d0)
           V=(QQ-F(INDEX,1))/.05d0*(F(INDEX+1,2)-F(INDEX,2))
     &     +F(INDEX,2)
           V=V*VCAP
520         IF(V.GT.0.01D0) THEN
                 XNUM=XA/V  ! number of time steps it takes flow
to move through pipe in 5 minutes
            ELSE
                XNUM=0.d0
            END IF
           NUM=INT(XNUM)!number of time steps it takes flow to
move through pipe in 5 minutes
           NZUM=NUM
           DA=XNUM-NZUM !difference to account for any
fractional differences in timestep
           DB=1.d0-DA
           !print*,'da, db, xnum, in(xnum)',DA,DB,XNUM,NUM
           II=I+NUM+1 !timestep at which inflow corresponding to
timestep (I) is leaving the pipe as outflow at timestep (II)
           !PRINT*,'XNUM',XNUM
           ! print*,'V',V
           !print*,'XA',XA
           !print*,'num',NUM
           !print*,'II',II
           !print*,'DA',DA
           !print*,'Q',Q
C -----------------------------------------------------------
C STORE INITIAL STORAGE CONDITIONS FOR PIPE FLOW AT 24 HOURS
C------------------------------------------------------------
            IF(I.EQ.288) THEN
                 PSTORE(inode)=STORE
                DAYDS(JDAY,INODE)=STORE*1233.48d0 !convert Acre-
Ft to cubic meters
            ENDIF
C -----------------------------------------------------------
           IF(II.GT.(5555-1))GO TO 522
           B(II)=B(II)+DA*Q
           II=I+NUM
           B(II)=B(II)+DB*Q
           !IF(TIME.LT.TIME1.OR.TIME.GT.TIME2)GO TO 550
           !WRITE(NUT,921)TIME,A(I),V,B(I),STORE
522      IF(TIME.GE.TIME1.AND.TIME.LE.TIME2)WRITE(NUT,921)TIME,
     &        A(I),V,B(I),STORE
550    CONTINUE
```

```
1000  B(5555)=I
C     CHECK Next line WITH THE BOOK, MIGUEL CAMPO CHANGE THE
LIMIT TO B(5555)-1
C     BECAUSE IT IS WRITING EXTRANGES PEAKS AFTER THE HYDROGRAPH
FINISH
       DO 1100 I=1,5555
           A(I)=B(I)
1100    CONTINUE
        DO 1200 I=2,288
          SUMQO24(I)=SUMQO24(I-1)+0.5d0*(A(I-1)+
     &        A(I))*5.d0*60.d0*0.0283168d0
1200    CONTINUE
          DAYQO(JDAY,INODE)=SUMQO24(288)
          DAYDS(JDAY,INODE)=SUMQI24(288)-SUMQO24(288)
       CALL MWRITE(NA,A)
        !ELSE
        !  WRITE(nut,999)
        !END IF
C -----------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C -----------------------------------------------------------------
----------
901   FORMAT(/,11X,'MODEL PIPEFLOW ROUTING OF STREAM',I2,'
WHERE',
     C /,11X,'STORAGE EFFECTS ARE NEGLECTED WITHIN THE PIPE,
FLOW',/,
     C 11X,'VELOCITIES ARE ESTIMATED BY ASSUMING STEADY FLOW
FOR',/,
     C 11X,'EACH UNIT INTERVAL(NORMAL DEPTH), AND FLOWS IN
EXCESS',/,
     C 11X,'OF (.82)(DIAMETER) ARE PONDED AT THE UPSTREAM
INLET:',/)

306   FORMAT(20X,'PIPELENGTH(FT) = ',F18.2,/,
     C 20X,'MANNINGS FACTOR = ',F17.3,/,
     C 20X,'UPSTREAM ELEVATION(FT) = ',F10.2,5X,/,
     C 20X,'DOWNSTREAM ELEVATION(FT) = ',F8.2,/,
     C 20X,'PIPE DIAMETER(FT) = ',F15.2,/)

905   FORMAT(/,13X,'NORMAL DEPTH VELOCITY PIPE ROUTING
RESULTS:',//,
     C 10X,'  TIME     INFLOW   VELOCITY   OUTFLOW
UPSTREAM',/,
     C 10X,' (HRS)     (CFS)     (FPS)      (CFS)
PONDING(AF)')
```

```
921   FORMAT(10X,F7.3,3F10.2,F13.3)

999   FORMAT(10X,'NO FLOW OCCURRED ON THIS DAY')

      RETURN
      END SUBROUTINE PIPER
```

```
C     PROGRAM 18  ! Based on Hromadka book pag 222
C --------------------------------------------------------------
-----------
      SUBROUTINE PRNODE
C --------------------------------------------------------------
-----------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCC
C THIS SUBROUTINE PRINTS A HYDROGRAPH AT THE SPECIFIED NODE
C
C VARIABLES:
C
C NUMA:  Stream1 to be graphed
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCC
C --------------------------------------------------------------
-----------
C   DECLARE VARIABLES
C   mref: number of unit hydrograph steps
C   nref: number of excess hyetograph steps
C --------------------------------------------------------------
------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/INPUTS/DATAINP(100,100)
      DIMENSION A(5555)
!     EXPORT Hydrograph, Date (hours) StreamA(CFS)
C --------------------------------------------------------------
--------------
C INPUT DATA
C --------------------------------------------------------------
--------------
      NA=INT(DATAINP(JCOUNT,4))
      TIME1=DATAINP(JCOUNT,5)
      TIME2=DATAINP(JCOUNT,6)
C --------------------------------------------------------------
--------------
C CALL STREAM STORAGE MATRIX
C --------------------------------------------------------------
--------------
      CALL MREAD(NA,A) ! reads columns from temporary storage
matrix SS
c      NUMBH=INT(H(5555)) ! converts to integers
```

```
C ------------------------------------------------------------------
-----------------
C OUTPUT - FORMAT
C ------------------------------------------------------------------
-----------------
      WRITE(NUT,101)NA
101   FORMAT(10X,'HYDROGRAPH OF STREAM NUMBER',I2)
C ------------------------------------------------------------------
-------------------------
C Print hydrograph (units in CFS)
C ------------------------------------------------------------------
-------------------------
      TIME=0.d0
      QIN=A(1)
      DO 200 I=1,5555
         TIME=TIME+.083333d0

IF(TIME.GE.TIME1.AND.TIME.LE.TIME2)WRITE(NUT,913)TIME,A(I)

C     C    B(I)
200   CONTINUE

913   FORMAT(10X,F7.3,1X,2F12.1)

      RETURN
      END SUBROUTINE PRNODE
```

```
C --------------------------------------------------------------
----------------------------------------------------------------
--------------------------------------------------------
C   Program: SCS-TR55 peak flow calculationn
C --------------------------------------------------------------
----------------------------------------------------------------
--------------------------------------------------
       SUBROUTINE q_peak(Area,Q,xIa,P,tc,j,qp,tp)
C --------------------------------------------------------------
-------------
C      version 3.0.1, Last ModIFied: See ModIFications below
C      WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C      Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons,
BAE (jep)
C                  University of Florida        BAE, NC State
University
C                  Gainesville, FL 32611        Raleigh, NC
27695-7625(USA)
C                  e-mail: carpena@ufl.edu
C --------------------------------------------------------------
-------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
-------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      DIMENSION ci(3,4,5)
C --------------------------------------------------------------
-------------
     DATA ci/68.0317,-82.907,11.1619,144.547,-130.64,-55.230,-
11.312,
     C 16.6125,-43.015,-11.505,-64.177,65.9007,-74.693,105.222,-
26.314,
     C -136.68,134.907,47.9565,12.1681,-
16.337,50.4334,14.2182,85.7116,
     C -85.806,24.9255,-42.167,16.1126,41.8526,-45.773,-13.503,-
6.5688,
     C 6.4981,-19.740,-7.8919,-38.206,39.0036,-3.9797,6.7479,-
2.9776,
     C -6.2829,6.585,2.1954,1.0577,-
1.1784,3.2996,1.3836,6.7419,-6.8946,
     C 2.5222,-0.8657,0.0456,2.3645,-0.6384,-0.2644,2.5021,-
0.5476,
     C -0.3427,2.4007,-0.8899,0.2078/
C --------------------------------------------------------------
-------------
c    DO 10 j=1,4
```

```
c       DO 10 i=1,3
c          WRITE(20,100) (ci(i,j,k),k=1,5)
c         WRITE(NUT,100) (ci(i,j,k),k=1,5)
c10     CONTINUE

C ---rmc 04/20/03 - Fix for Q=0
      IF(Q.le.0) THEN
         qp=0.d0
        tp=qp
         RETURN
      END IF
C ---rmc 04/20/03 - END of fix for Q=0
     xIaP=xIa/P
C --- TR55 stablishes that IF Ia/P is outside the range
(0.1<Ia/P<0.5),
C ----use the limiting values
     IF(xIa/P.gt.0.5d0)xIaP=0.5d0
     IF(xIa/P.lt.0.1d0)xIaP=0.1d0

C -- Import Ia/P and storm type I,IA,II,III (j=1,4) ------------
     C0=ci(1,j,1)*xIaP**4+ci(1,j,2)*xIaP**3+ci(1,j,3)*xIaP**2+
     C     ci(1,j,4)*xIaP+ci(1,j,5)
     C1=ci(2,j,1)*xIaP**4+ci(2,j,2)*xIaP**3+ci(2,j,3)*xIaP**2+
     C     ci(2,j,4)*xIaP+ci(2,j,5)
     C2=ci(3,j,1)*xIaP**4+ci(3,j,2)*xIaP**3+ci(3,j,3)*xIaP**2+
     C     ci(3,j,4)*xIaP+ci(3,j,5)

C --- Unit q peak, qp (m3/s) -----------------------------------
     qu=4.3046d0*10.d0**(C0+C1*dlog10(tc)+C2*(dlog10(tc))**2-
6.d0)
     Fp=0.75d0
     qp=qu*Area*Q*Fp

C -- Unit hydrograph time to peak (min) ------------------------
     tp=0.127481d0*Q*Area/qp/60.d0
C -------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C -------------------------------------------------------------
----------
100   FORMAT(5f9.3)

     RETURN
     END SUBROUTINE q_peak
```

```
!  ReadData.F
!
!  FUNCTIONS:
!  Console1 - Entry point of console application.
!

!****************************************************************
*************
!
! SUBROUTINE: ReadData
!
!  PURPOSE:  Entry point for the console application.
!
!****************************************************************
*************


      SUBROUTINE READDATA(NRNODES,NRDAYS,FIRR,DAREA)

      implicit double precision (a-h,o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)

common/CINPUT/DETO(5555,100),DBF(5555,100),DTAVG(5555,100),
     &
DTMAX(5555,100),DTMIN(5555,100),DWS2(5555,100),DSORAD(5555,100),
     &
DCKM(5555,100),DAB(5555,100),DIRR(5555,100),DSNO(5555,100)
      COMMON/INPUTS/DATAINP(100,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE

COMMON/NNINOUT/NIET,NIBF,NITM,NITU,NITL,NIWS,NISR,NICK,NIAB,NIIR
,
     &
NOPR,NORO,NOET,NODN,NOSM,NOQI,NODS,NOAB,NOIR,NOQO,NOMO,NOPE,NOWB
,
     & NOSD,NISN,NOIN
      Dimension NINP(13),FIRR(1,100),DAREA(100)
      DATA NINP/19,6,8,9,17,2,4,6,5,5,25,27,3/

!   INITIALIZE ARRAY TO ZERO
      DO 5 ICOUNT = 1,100
          DO 5 J = 1,100
          DATAINP(ICOUNT,J)=0.D0
5     CONTINUE
```

```
      DO 10 J = 1,100
          FIRR(1,J)=0.D0
          DAREA(J)=0.D0
10    CONTINUE

!    LOOP TO READ INPUTS AND STORE THEM INTO DATA MATRIX
      KODE = 0
      ICOUNT=0
      DO WHILE (KODE.NE.999)
          !print*,'start kode', KODE
          ICOUNT = ICOUNT+1
          READ(NDAT,*)(DATAINP(ICOUNT,J), J=1,3)   !read NZ1,
NZ2, KODE
c          print*,'inputs',icount,(DATAINP(ICOUNT,J), J=1,3)
          KODE = INT(DATAINP(ICOUNT,3))
          IF (KODE.NE.999) THEN
              !print*,'icount,kode',icount,kode
              JMAX = 3+NINP(DATAINP(ICOUNT,3))
              READ(NDAT,*)(DATAINP(ICOUNT,J),J=4,JMAX)
              IF (KODE.EQ.3) THEN
                  JMAX = 10+3*(DATAINP(ICOUNT,8))
                  READ(NDAT,*)(DATAINP(ICOUNT,J),J=11,JMAX)
              END IF
              !print*,int(JMAX)

!WRITE(nowb,910),icount,(DATAINP(ICOUNT,J),J=1,JMAX)
          END IF

      END DO
c---------SKIPPING FIRST TWO LINES IN INPUT FILES
      READ(NIET,*)
      READ(NIBF,*)
      READ(NITM,*)
      READ(NITU,*)
      READ(NITL,*)
      READ(NIWS,*)
      READ(NISR,*)
      READ(NICK,*)
      READ(NIAB,*)
      READ(NIIR,*)
      READ(NISN,*)
      READ(NIET,*)
      READ(NIBF,*)
      READ(NIBF,*)
      READ(NITM,*)
      READ(NITU,*)
      READ(NITL,*)
```

```
      READ(NIWS,*)
      READ(NISR,*)
      READ(NICK,*)
      READ(NIAB,*)
      READ(NIIR,*)
      READ(NISN,*)

!     READ FRACTION OF WATERSHED IRRIGATED (FIRR)
      READ(NIIR,*)N,(FIRR(1,K),K=1,NRNODES)
!     READ WATERSHED AREAS (DAREA)
      READ(NIIR,*)N,(DAREA(K),K=1,NRNODES)
      !PRINT*,'DAREA',(DAREA(K),K=1,NRNODES)


!     LOOP TO READ HYDROLOGIC INPUT MATRICES
      DO 100 I=1,NRDAYS+1
          READ(NIPR,*)N,(DPRECIP(I,K),K=1,NRNODES)
          READ(NIET,*)N,(DETO(I,K),K=1,NRNODES)
          READ(NIBF,*)N,(DBF(I,K),K=1,NRNODES)
          READ(NITM,*)N,(DTAVG(I,K),K=1,NRNODES)
          READ(NITU,*)N,(DTMAX(I,K),K=1,NRNODES)
          READ(NITL,*)N,(DTMIN(I,K),K=1,NRNODES)
          READ(NIWS,*)N,(DWS2(I,K),K=1,NRNODES)
          READ(NISR,*)N,(DSORAD(I,K),K=1,NRNODES)
          READ(NICK,*)N,(DCKM(I,K),K=1,NRNODES)
          READ(NIAB,*)N,(DAB(I,K),K=1,NRNODES)
          READ(NIIR,*)N,(DIRR(I,K),K=1,NRNODES)
          READ(NISN,*)N,(DSNO(I,K),K=1,NRNODES)
100   CONTINUE

C ----------------------------------------------------------------
----------
C  ADD HEADERS TO DAILY OUTPUT FILES
C ----------------------------------------------------------------
----------
      WRITE(NOPR,902)(INT(DPRECIP(1,K)),K=1,NRNODES) !writes the
node headers
      WRITE(NORO,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NOET,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NODN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NOSM,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NOQI,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NODS,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NOAB,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NOIR,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NDSS,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NOQO,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
```

```
      WRITE(NOMO,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NOPE,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      !WRITE(NOWB,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      WRITE(NOSD,902)(INT(DPRECIP(1,K)),K=1,NRNODES)

C --------------------------------------------------------------
----------
C  WRITE ALL DATA TO FILES THAT ARE THE SAME AS INPUT FILES
C --------------------------------------------------------------
----------
      DO 200 I=1,NRDAYS
          WRITE(NOPR,910)I,(DPRECIP(I+1,K),K=1,NRNODES)
!         WRITE(NOET,910)I,(DETO(I+1,K),K=1,NRNODES)
!          WRITE(NODN,910)I,(DBF(I+1,K),K=1,NRNODES) lw NEED TO
MAKE ANOTHER OUTPUT FILE FOR BASEFLOW??
!          WRITE(NOSM,910)I,(DSM(I+1,K),K=1,NRNODES)
          WRITE(NOAB,910)I,(DAB(I+1,K),K=1,NRNODES)
          WRITE(NOIR,910)I,(DIRR(I+1,K),K=1,NRNODES)
200   CONTINUE
C --------------------------------------------------------------
----------
C  WRITE ALL INPUT FILES TO .OINP FILE (NOIN)
C --------------------------------------------------------------
----------
C ---precipitation
      WRITE(NOIN,911)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 210 I=1,NRDAYS
          WRITE(NOIN,910)I,(DPRECIP(I+1,K),K=1,NRNODES)
210   CONTINUE
C ---potential ET
      WRITE(NOIN,912)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 211 I=1,NRDAYS
          WRITE(NOIN,910)I,(DETO(I+1,K),K=1,NRNODES)
211   CONTINUE
C ---BASEFLOW
      WRITE(NOIN,913)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 212 I=1,NRDAYS
          WRITE(NOIN,910)I,(DBF(I+1,K),K=1,NRNODES)
212   CONTINUE
C ---AVERAGE TEMP
      WRITE(NOIN,914)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 213 I=1,NRDAYS
          WRITE(NOIN,910)I,(DTAVG(I+1,K),K=1,NRNODES)
```

```
213   CONTINUE
C ---MAX TEMP
      WRITE(NOIN,915)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 214 I=1,NRDAYS
          WRITE(NOIN,910)I,(DTMAX(I+1,K),K=1,NRNODES)
214   CONTINUE
C ---MIN TEMP
      WRITE(NOIN,916)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 215 I=1,NRDAYS
          WRITE(NOIN,910)I,(DTMIN(I+1,K),K=1,NRNODES)
215   CONTINUE
C ---WIND SPEED
      WRITE(NOIN,917)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 216 I=1,NRDAYS
          WRITE(NOIN,910)I,(DWS2(I+1,K),K=1,NRNODES)
216   CONTINUE
C ---SOLAR RADIATION
      WRITE(NOIN,918)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 217 I=1,NRDAYS
          WRITE(NOIN,910)I,(DSORAD(I+1,K),K=1,NRNODES)
217   CONTINUE
C ---CROP COEFFICIENT
      WRITE(NOIN,919)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 218 I=1,NRDAYS
          WRITE(NOIN,910)I,(DCKM(I+1,K),K=1,NRNODES)
218   CONTINUE
C ---WATER ABSTRACTION
      WRITE(NOIN,920)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 219 I=1,NRDAYS
          WRITE(NOIN,910)I,(DAB(I+1,K),K=1,NRNODES)
219   CONTINUE
C ---IRRIGATION
      WRITE(NOIN,921)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 220 I=1,NRDAYS
          WRITE(NOIN,910)I,(DIRR(I+1,K),K=1,NRNODES)
220   CONTINUE
C ---SNOWMELT
      WRITE(NOIN,922)
      WRITE(NOIN,902)(INT(DPRECIP(1,K)),K=1,NRNODES)
      DO 221 I=1,NRDAYS
```

```
           WRITE(NOIN,910)I,(DSNO(I+1,K),K=1,NRNODES)
221    CONTINUE



!902    FORMAT(3X,'DAY',100(7X,'(',I3,')'))
902    FORMAT(3X,'DAY',100(9X,I3))
910    FORMAT(I5,3X,100E12.5)
911    FORMAT(17X,'>>>>> PRECIPITATION, MM <<<<<')
912    FORMAT(17X,'>>>>> POTENTIAL ET, MM <<<<<')
913    FORMAT(17X,'>>>>> BASEFLOW, m3/s <<<<<')
914    FORMAT(17X,'>>>>> AVERAGE TEMPERATURE, C <<<<<')
915    FORMAT(17X,'>>>>> MAXIMUM TEMPERATURE, C <<<<<')
916    FORMAT(17X,'>>>>> MINIMUM TEMPERATURE, C <<<<<')
917    FORMAT(17X,'>>>>> WIND SPEED, CM/S <<<<<')
918    FORMAT(17X,'>>>>> SOLAR RADIATION, <<<<<')
919    FORMAT(17X,'>>>>> CROP COEFFICIENT, Kcmid <<<<<')
920    FORMAT(17X,'>>>>> WATER ABSTRACTION, MM <<<<<')
921    FORMAT(17X,'>>>>> IRRIGATION, MM <<<<<')
922    FORMAT(17X,'>>>>> SNOWMELT, M3/S <<<<<')

       RETURN
       END
```

```
C --------------------------------------------------------------
---------------------
C   Program: Output summary of hydrology results nicely
C --------------------------------------------------------------
---------------------
      SUBROUTINE results(P,CN,Q,Area,tc,xIa,jstype,D,pL,Y,
     &  ieroty)
C --------------------------------------------------------------
---------------------
C  version 3.0.1, Last ModIFied: See ModIFications below
C  WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C  Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons, BAE
(jep)
C            University of Florida        BAE, NC State
University
C            Gainesville, FL 32611        Raleigh, NC 27695-
7625(USA)
C           e-mail: carpena@ufl.edu
C --------------------------------------------------------------
---------------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
---------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      CHARACTER*4 stype(5)
      DATA stype/'I','IA','II','III','user'/
C --------------------------------------------------------------
---------------------
      WRITE(NUT,99)
      WRITE(NUT,999)
      WRITE(NUT,*)'   INPUTS:'
      WRITE(NUT,999)
      WRITE(NUT,100)P
      WRITE(NUT,200)stype(jstype)
      WRITE(NUT,250)D
      WRITE(NUT,300)CN
      WRITE(NUT,400)Area
      WRITE(NUT,500)pL
      WRITE(NUT,600)Y*100.d0
C --------------------------------------------------------------
--------------------
      IF (ieroty.eq.1)THEN       !! 1) Williams (1975)
         WRITE(NUT,610) ieroty,'; Williams (1975)'
      ELSE IF (ieroty.eq.2)THEN !! 2) GLEAMS/daily CREAMS
```

```
         WRITE(NUT,610) ieroty,'; GLEAMS'
      ELSE IF (ieroty.eq.3)THEN !! 3) Foster et al. (1977)
         WRITE(NUT,610) ieroty,'; Foster et al., (1997)'
      ELSE IF (ieroty.eq.4)THEN !! 4) Cooley (1980) - Design
Storm
         WRITE(NUT,610) ieroty,'; Cooley (1980)'
      ELSE
         WRITE(NUT,*)'ERROR: Incorrect erosion type (1-4)'
         STOP
      END IF
C ---------------------------------------------------------------
----------
      WRITE(NUT,999)
      WRITE(NUT,*)'   OUTPUTS:'
      WRITE(NUT,999)
      WRITE(NUT,700)Q,Q*Area*10.d0
      WRITE(NUT,800)xIa
      WRITE(NUT,900)tc,tc*60.d0
      WRITE(NUT,*)' '
C ---------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C ---------------------------------------------------------------
----------
99    FORMAT(10x,
     C'>>> HYDROGRAPH CALCULATION FOR WATERSHED-NRCS METHOD
<<<')
100   FORMAT(4x,'Storm Rainfall    =  ',f8.2,' mm')
200   FORMAT(4x,'SCS Storm Type    =     ',a4,'  type')
250   FORMAT(4x,'Storm Duration    = ',f8.1,'  hr')
300   FORMAT(4x,'NRCS Curve Number = ',f8.1)
400   FORMAT(4x,'Watershed Area    =  ',f8.2,' ha')
500   FORMAT(4x,'Maximum Flow Path Length   =',f8.2,' m')
600   FORMAT(4x,'Average slope of flow path =',f8.2,' %')
610   FORMAT(4x,'Erosion MUSLE type = ',i3,A25,',(See Manual)')
700   FORMAT(4x,'Runoff Volume       =',f15.2,' mm =',f15.2,'
m3')
800   FORMAT(4x,'Initial Abstraction =',f8.2,' mm')
900   FORMAT(4x,'Concentration Time  =',f8.2,' hr =',f8.2,'
min')
999   FORMAT(2x,18('-'))

      RETURN
      END SUBROUTINE results
```

```
C --------------------------------------------------------------
---------------------
C   Program:
C --------------------------------------------------------------
---------------------
      FUNCTION SCStorm(Jstype,ptime)
C --------------------------------------------------------------
---------------------
c ! SCS design storm type equation using generalized
coefficients
c ! (Munoz-Carpena and Parsons,2004) for 24 hour storms,
c !     P(t)       t-b (    d    ) ^g
c !     ---- = a + ----(---------)
c !     P24         c  (e|t-b|+f )
C --------------------------------------------------------------
---------------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
---------------------
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
C      PARAMETER (5555=INT(600))

COMMON/rain/rfix,rti(5000),rfi(5000),rcum(5000,2),ref(5000,2),nc
um
      DIMENSION cff(4,7)
C --------------------------------------------------------------
---------------------
      DATA
cff/0.4511d0,0.3919d0,0.495d0,0.5d0,9.995d0,7.96d0,11.8d0,
     C   12.d0,1d0,1.d0,0.56d0,24.d0,-
0.1617d0,0.843d0,10.6d0,24.04d0,
     C   -
3.0163d0,120.39d0,130.d0,2.d0,0.013d0,0.3567d0,0.525d0,
     C   0.04d0,0.5853d0,0.4228d0,0.75d0,0.75d0/

c     DO 10 i=1,4
c        WRITE(*,100)(cff(i,j), j=1,7)
c        WRITE(NUT,100)(cff(i,j), j=1,7)
c10   CONTINUE

      IF(Jstype.le.4) THEN
         cffa=cff(Jstype,1)
         cffb=cff(Jstype,2)
         cffc=cff(Jstype,3)
         cffd=cff(Jstype,4)
         cffe=cff(Jstype,5)
```

```
         cfff=cff(Jstype,6)
         cffg=cff(Jstype,7)
         bigT= ptime-cffb
         denom=cffe*dabs(bigT)+cfff
         IF(Jstype.eq.1.and.denom.ge.0.d0) THEN
            SCStorm=0.5129d0
         ELSE
            SCStorm=cffa+(bigT/cffc)*(cffd/denom)**cffg
         END IF
      ELSE
         DO 15 i=2,ncum
            t1=rcum(i,1)
            rcum1=rcum(i,2)
            t2=rcum(i+1,1)
            rcum2=rcum(i+1,2)
            IF(ptime.gt.t1.and.ptime.le.t2) THEN
            SCStorm=(ptime-t1)/(t2-t1)*(rcum2-
     C      rcum1)+rcum1
            END IF
15       CONTINUE
      END IF

c100  FORMAT(7f9.4)

       RETURN
       END FUNCTION SCStorm
C ---------------------------------------------------------------
---------------------
      SUBROUTINE runoff(P,CN,xIa,Q)
C ---------------------------------------------------------------
---------------------
C SCS runoff calculation
C ---------------------------------------------------------------
---------------------
C      version 3.0.1, Last ModIFied: See ModIFications below
C      WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C      Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons,
BAE (jep)
C                  University of Florida        BAE, NC State
University
C                  Gainesville, FL 32611        Raleigh, NC
27695-7625(USA)
C                  e-mail: carpena@ufl.edu
C ---------------------------------------------------------------
---------------------
C      DEFINE VARIABLES
```

```
C --------------------------------------------------------------
---------------------
C      P: PRECIPITATION, mm
C      CN: CURVE NUMBER, dimensionless
C      xIa: INITIAL ABSTRACTION, mm
C      Q: RUNOFF TOTAL, mm

C      DECLARE VARIABLES
C --------------------------------------------------------------
---------------------
     IMPLICIT DOUBLE PRECISION (a-h, o-z)
C --------------------------------------------------------------
---------------------
     S=25400.d0/CN-254.d0
     S20=0.2d0*S
c ---- modified 7.17.2023 llw -----
      S05=1.33d0*(S20**1.15d0)
      xIa=0.05*S05
      IF(P.gt.0.05d0*S05)THEN
         Q=(P-0.05d0*S05)**2.d0/(P+0.95d0*S05)
      ELSE
         Q=0.0d0
      END IF
      !PRINT*,'S,S20,S05,XIA,Q',S,S20,S05,XIA,Q

     RETURN
      END SUBROUTINE runoff
```

```
C ----------------------------------------------------------------
---------------------
C PROGRAM 17  - Based on Hromadka book pag 217
C ----------------------------------------------------------------
---------------------
      SUBROUTINE SEE (X,D1,D2,I1,I2,NUT,Y,NB,TIME)
C ----------------------------------------------------------------
---------------------
C   DECLARE VARIABLES
C ----------------------------------------------------------------
---------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      DIMENSION Y(20)
C ----------------------------------------------------------------
---------------------
      DO 100 K=2,NB
         IF(X.LT.Y(K))GO TO 200
100   CONTINUE
      TI=TIME+.083333d0
C ----------------------------------------------------------------
---------------------
      WRITE(NUT,101)TI
      K=NB
200   I1=K-1
      I2=K
      D1=Y(I1)
      D2=Y(I2)
C ----------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C ----------------------------------------------------------------
----------
101   FORMAT(10X,F7.3,5X,
     C  '*BASIN CAPACITY EXCEEDED: BASIN DATA IS EXTRAPOLATED*')
c rmc 1000    CONTINUE
      RETURN
      END SUBROUTINE SEE
```

```
C -----------------------------------------------------------------
--------------
!     PROGRAM 18  -  Based on Hromadka book pag 222
C -----------------------------------------------------------------
--------------
      SUBROUTINE SPLIT(DAYQI,DAYQO,DAYMO)
C -----------------------------------------------------------------
--------------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCC
C THIS SUBROUTINE SPLITS STREAM "A" INTO STREAM "A" AND STREAM
"B"           C
C VARIABLES:
C
C NA:  Stream "A" number. This stream is the one to be modeled
C
C NB:  Stream "B" number [0 for moving the excess flow from
stream "A"       C
C      to a permanent storage; 1 for moving excess flow from
C
C       stream "A" to stream "B"].
                C
C PB:  Percentage (decimal) of stream to be diverted
C
C TIME1:  Time for Beginning of results (hrs)
C
C TIME2:  Time for End of results (hrs)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCC
C -----------------------------------------------------------------
--------------
C   DECLARE VARIABLES
C -----------------------------------------------------------------
--------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      DIMENSION A(5555),B(5555),DAYQI(5555,100),DAYQO(5555,100),
     & DAYMO(5555,100),SUMA24(5555),SUMB24(5555)
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)
      COMMON/INPUTS/DATAINP(100,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      DIMENSION SUMQI24(5555),SUMQO24(5555)
C -----------------------------------------------------------------
--------------
```

```
C   INITIALIZE VARIABLES
C --------------------------------------------------------------
--------------
      TIME=0.d0
C --------------------------------------------------------------
--------------
C   READ INPUT DATA
C --------------------------------------------------------------
--------------
      NA=INT(DATAINP(JCOUNT,4))
      NB=INT(DATAINP(JCOUNT,5))
      PB=DATAINP(JCOUNT,6)
      TIME1=DATAINP(JCOUNT,7)
      TIME2=DATAINP(JCOUNT,8)
      AREA=DATAINP(JCOUNT,9)
      NOUT1=INT(TIME1*12.d0+.01d0)
      NOUT2=INT(TIME2*12.d0+.01d0)
      DO 5 J=1,100
            IF (INT(DPRECIP(1,J)).EQ.NZ2) THEN
               INODE=J
            END IF
5     CONTINUE
      SUMA24=0.D0
      SUMB24=0.D0
C --------------------------------------------------------------
--------------
      WRITE(NUT,901)NA,NB
C --------------------------------------------------------------
--------------
C   GRAPHICS
C --------------------------------------------------------------
--------------
      PA=1.d0-PB
C --------------------------------------------------------------
--------------
      WRITE(NUT,921)NB,NA,PB,NA
      WRITE(NUT,923)NB,PA,NA,PB,NA
      WRITE(NUT,903)NA,NB,PA,NA,PB,NB
      WRITE(NUT,905)NB,NA,NB,NA
C --------------------------------------------------------------
--------------
C READ IN STREAM DATA
C --------------------------------------------------------------
--------------
      CALL MREAD(NA,A)
      CALL MREAD(NB,B)
```

```
C --- Calculate and write inflow volume (in mm) to storage
matrix
      SUMA24(1)=0.5d0*A(1)*5.d0*60.d0*0.0283168d0
      SUMB24(1)=0.5d0*B(1)*5.d0*60.d0*0.0283168d0
      SUMQI24(1)=SUMA24(1)+SUMB24(1)
      DO 10 I=2,288
          SUMA24(I)=SUMA24(I-1)+0.5d0*(A(I-1)+
     &                A(I))*5.d0*60.d0*0.0283168d0 !Calculating
the sum by taking average of timesteps, multiplying over time
step to get volume, and converting to cubic meters
          SUMB24(I)=SUMB24(I-1)+0.5d0*(B(I-1)+
     &                B(I))*5.d0*60.d0*0.0283168d0
          SUMQI24(I)=SUMA24(I)+SUMB24(I)
10    CONTINUE

      DAYQI(JDAY,INODE)=SUMQI24(288) ! Daily total inflow volume
is equal to sum at timestep 288

C ------------------------------------------------------------
--------------
C   MODEL SPLITFLOW
C ------------------------------------------------------------
--------------
      NUMBER=INT(A(5555))
      !IF (NUMBER.GT.0.D0) THEN
       IF(PB.EQ.0.d0)GO TO 1000
       DO 100 I=1,NUMBER
           X=PB*A(I)
           AIN=A(I)
           BIN=B(I)
           B(I)=B(I)+X
           A(I)=A(I)-X
           TIME=TIME+.08333d0
           IF(I.LT.NOUT1.OR.I.GT.NOUT2)GO TO 100
C ------------------------------------------------------------
--------------
               WRITE(NUT,906)TIME,BIN,AIN,B(I),A(I)
100      CONTINUE
1000  CONTINUE
       NUMB=INT(B(5555))
       IF(NUMBER.GT.NUMB)NUMB=NUMBER
           B(5555)=NUMB
       CALL MWRITE(NA,A)
        SUMQO24(1)=0.5d0*B(1)*5.d0*60.d0*0.0283168d0
       DO 1100 I=1,5555
               A(I)=B(I)
          IF (I.GT.1) THEN
```

```
          SUMQO24(I)=SUMQO24(I-1)+0.5d0*(A(I-1)+
     &                A(I))*5.d0*60.d0*0.0283168d0
          END IF
1100  CONTINUE
       CALL MWRITE(NB,B)
          DAYQO(JDAY,INODE)=SUMQO24(288)
          DAYMO(JDAY,INODE)=SUMQI24(288)-SUMQO24(288)
      !ELSE
      !    WRITE(nut,999)
      !END IF
C --------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C --------------------------------------------------------------
----------
901   FORMAT(/,10X,'MODEL STREAM SPLITFLOW WHERE A CONSTANT
PROPORTION'
     C ,/,10X,'OF STREAM',I2,' IS ADDED TO STREAM',I2,' :',//)
921   FORMAT(
     C 20X,'    INFLOW               INFLOW',/,
     C 20X,'  (STREAM:',I2,')          (STREAM:',I2,')' ,/,
     C 27X,'|',19X,'|',/,27X,'|',19X,'|',/,
     C 27X,'| (',F4.3,')(STREAM:',I2,') |')
923   FORMAT (
     C 27X,'|<------------------* < = splitFLOW Model',/,
     C 27X,'|',19X,'|',/,27X,'|',19X,'|',/,
     C 27X,'|',19X,'|',/,
     C
27X,'V',19X,'V',/,24X,'STREAM:',I2,12X,'(',F6.3,')(STREAM:',I2,
     C')',/,20X,'+ (',F6.3,')(STREAM:',I2,')',//)
903   FORMAT(11x,'STREAM NUMBER:',I2,' IS SPLIT TOWARDS
STREAM:',I2,/,
     C 11X,'WHERE',F6.2,' (DECIMAL PERCENT) REMAINS IN STREAM:
',I2,/,
     C 11X,'AND ',F6.2,'  (DECIMAL PERCENT) IS ADDED TO
STREAM:',I2)
905   FORMAT(//,22X,'STREAM SPLITFLOW MODELING RESULTS:',//,
     C 11X,' MODEL    INFLOW      INFLOW      OUTFLOW
OUTFLOW',/,
     C 11X,' TIME    STREAM', I2,3 (4X,  'STREAM',I2),/,
     C 11X,' (HRS)     (CFS)       (CFS)      (CFS)
(CFS)')
906   FORMAT(10X,F7.3,3X,4(F8.1,4X))
999   FORMAT(10X,'NO FLOW OCCURRED ON THIS DAY')
C --------------------------------------------------------------
--------------
```

```
RETURN
END SUBROUTINE SPLIT
```

```
C  -------------------------------------------------------------
----------
C    PROGRAM 13 - Based on Hromadka book pag 164
C  -------------------------------------------------------------
----------
C   THIS SUBROUTINE DETERMINES THE PERCENTAGES OF DISCHARGE
C   FACTORS FOR THE VARIOUS ZONE CLASSIFICATIONS.
C   RFAT VMNT
C --------------------------------------------------------------
-----------
!      SUBROUTINE   SUBSB(TIMLAG,PERCNT,KODE1,NUMBER,NUT)
      SUBROUTINE   SUBSB(TIMLAG,PERCNT,KODE1,NUMBER)
C --------------------------------------------------------------
-----------
C   DECLARE VARIABLES
C --------------------------------------------------------------
-----------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      DIMENSION
PERCNT(150),VAL(31,2),FOOT(31,2),VMNT(33,2),SCS(33,2)
      DIMENSION DESERT(33,2)
      DATA VAL/0.,15.0,25.,35.0,50.,65.0,75.,100.,115.,125.,
     C
140.0,150.0,165.0,175.0,200.0,225.0,250.,275.,300.,325.,350.,
     C
375.,400.,450.,500.,550.,600.,650.,700.,750.,99999.,0.0,2.6,
     C
5.0,8.6,15.5,25.0,32.0,50.0,57.9,62.0,66.8,69.5,72.6,74.3,
     C 78.0,81.0,83.5,85.7,87.5,89.0,90.5,91.6,92.7,94.3,95.8,
     C 96.9,97.8,98.5,99.0,99.5,100. /
      DATA FOOT/0.,15.0,25.,35.0,50.,60.0,75.,85.0,90.0,95.0,
     C 100.0,110.0,125.0,140.0,150.0,175.0,200.0,225.0,250.0,
     C 0275.0,300.0,325.0,350.0,375.0,400.0,450.0,500.0,550.,0,
     C 600.0,650.0,99999.,0.0,1.9,3.8,6.0,10.3,14.0,21.7,29.0,
     C 34.2,43.3,50.0,56.9,63.8,69.0,71.9,77.8,82.4,86.0,89.0,
     C 91.4,93.4,95.0,96.2,97.2,97.9,98.5,99.0,99.3,99.7,
     C 99 9 100  /
      DATA vMNT/0.,15.0,25.,35.0,40.0,50.,65.0,75.0,90.0,
     C 100.0,115.0,125.0,140.0,150.0,175.0,200.0,225.0,250.0,
     C 275.0,300.0,325.0,350.0,375.0,400.0,450.0,500.0,
     C 550.0,600.0,650.0,700.0,750.0,800.0,99999.0,0.0,
     C 3.3,6.7,10.6,13.4,21.0,33.0,39.3,46.3,50.0,54.2,
     C 56.7,59.8,61.8,65.8,69.2,72.2,74.8,77.0,79.0,80.7,
     C 82.2,83.5,84.8,86.9,88.9,90.5,92.0,93.3,94.5,95.5,
     C 96.4,100./
      DATA DESERT/0.,12.5,25.0,37.5,50.0,62.5,75.0,87.5,100.,
```

```
     C
112.5,125.,137.5,150.,162.5,175.,187.5,200.,225.,250.,275.,
     C
300.,325.,350.,375.,400.,450.,500.,550.,600.,700.,800.,1000.,
     C
9999.,0.,1.1,3.2,6.3,10.5,18.5,31.3,42.0,50.0,56.5,61.3,65.2,
     C
68.5,71.5,74.0,76.2,78.3,81.6,84.3,86.7,88.7,90.2,91.6,92.8,
     C 93.9,95.6,96.9,97.8,98.3,99.5,99.9,99.99,100./
      DATA SCS/0.,8.6,17.2,25.9,34.5,43.1,51.7,60.3,69.,77.6,
     C 86.2,94.8,103.4,112.1,120.7,129.3,137.9,146.5,155.2,
     C 163.8,172 4,189.6,206.9,224 1,241.4,258.6,275.8,293.1,
     C 310.3,327 6,344.8,387.9,999.,0.,.1,.6,1.2,3.5,6.5,10.7,
     C 16.3,22.8,30.,37.5,45.,52.2,58.9,65.,70.,75.1,79.,82.2,
     C 84.9,87.1,90.8,93.4,95.3,96
7,97.7,98.4,98.9,99.3,99.5,99.7,
     C 99.9,100./
C -------------------------------------------------------------
-----------
c      KODE1=1:  VALLEY
c      KODE1=2:  FOOTHILL
c      KODE1=3:  MOUNTAIN
c      KODE1=4:  DESERT
c      KODE1=5:  NOT DIRECTLY USED. LINEAR INTERPOLATION OF 1-4
c      KODE1=6:  SCS
c      TIMLAG=LAG
c      TIME=INCREMEMTS OF LAG
c      NUMBER=INDEX OF PERCNT VECTOR
C -------------------------------------------------------------
-----------
      ANEW=0.d0
      AOLD=0.d0
      K=1
      NUMBER=1
      TIME=TIMLAG
10    K=K+1
      IF(NUMBER.GE.151)GO TO 1000
         N=K-1
         GO TO(100,200,300,350,355,355),KODE1
100   CONTINUE
C -------------------------------------------------------------
-----------
C   INTEGRATE "S" GRAPH IN ORDER TO DETERMINE UH(I)
C -------------------------------------------------------------
-----------
      TEMP=0.5d0*(VAL(K,2)+VAL(N,2))*
     C(VAL(K,1)-VAL(N,1))
```

```
      ANEW=ANEW+TEMP
      IF(TIME.GT.VAL(K,1))GO TO 10
         Y=VAL(K,2)
         X=VAL(N,2)
         DEL=(TIME-VAL(N,1))/(VAL(K,1)-VAL(N,1))
         B=VAL(K,1)-VAL(N,1)
         GO TO 400
200   CONTINUE
      TEMP=0.5d0*(FOOT(K,2)+FOOT(N,2))*
     C(FOOT(K,1)-FOOT(N,1))
      ANEW=ANEW+TEMP
      IF(TIME.GT.FOOT(K,1))GO TO 10
         Y=FOOT(K,2)
         X=FOOT(N,2)
         DEL=(TIME-FOOT(N,1))/(FOOT(K,1)-FOOT(N,1))
         B=FOOT(K,1)-FOOT(N,1)
         GO TO 400
300   CONTINUE
      TEMP=0.5d0*(VMNT(K,2)+VMNT(N,2))*
     C(VMNT(K,1)-VMNT(N,1))
      ANEW=ANEW+TEMP
      IF(TIME.GT.VMNT(K,1))GO TO 10
         Y=VMNT(K,2)
         X=VMNT(N,2)
         DEL=(TIME-VMNT(N,1))/(VMNT(K,1)-VMNT(N,1))
         B=VMNT(K,1)-VMNT(N,1)
         GO TO 400
350   CONTINUE
      TEMP=0.5d0*(DESERT(K,2)+DESERT(N,2))*
     C(DESERT(K,1)-DESERT(N,1))
      ANEW=ANEW+TEMP
      IF(TIME.GT.DESERT(K,1))GO TO 10
         Y=DESERT(K,2)
         X=DESERT(N,2)
         DEL=(TIME-DESERT(N,1))/(DESERT(K,1)-DESERT(N,1))
         B=DESERT(K,1)-DESERT(N,1)
         GO TO 400 !! not continue??
C  --------------------------------------------------------------
---------
C  SCS METHOD
C  --------------------------------------------------------------
---------
355   TEMP=0.5d0*(SCS(K,2)+SCS(N,2))*(SCS(K,1)-SCS(N,1))
      ANEW=ANEW+TEMP
C
      IF(TIME.GT.SCS(K,1))GO TO 10
         Y=SCS(K,2)
```

```
          X=SCS(N,2)
          DEL=(TIME-SCS(N,1))/(SCS(K,1)-SCS(N,1))
          B=SCS(K,1)-SCS(N,1)
C
400   CONTINUE
      DEL=DEL*(Y-X)
      XX=X+DEL
C  -------------------------------------------------------------
---------
C   ADJUST INTEGRATION FOR INTERPOLATION
C  -------------------------------------------------------------
---------
      DELA=0.5d0*(Y+XX)*(1.d0-DEL/(Y-X))*B
      ANEW=ANEW-DELA
      PERCNT(NUMBER)=(ANEW-AOLD)/TIMLAG
      NUMBER=NUMBER+1
      AOLD=ANEW
      ANEW=ANEW-.5d0*(X+XX)*DEL/(Y-X)*B
      TIME=TIME+TIMLAG
      K=K-1
      IF(NUMBER.EQ.2)GO TO 10
         DELX=PERCNT(NUMBER-1)-PERCNT(NUMBER-2)
         IF(DELX.LE..51d0)GO TO 1000
            GO TO 10
1000     CONTINUE
      NUMBER=NUMBER-1
C -------------------------------------------------------------
---------
      IF(NUMBER.GE.150)WRITE(NUT,1001)NUMBER
C  -------------------------------------------------------------
---------
C    FINISH-OFF HYDROGRAPH
C  -------------------------------------------------------------
---------
      IF(NUMBER.GE.150)GO TO 1250
         NNUM=150-NUMBER
         XNUM=NNUM
         REM=100.d0-PERCNT(NUMBER)
         REM1=REM/XNUM
         DELX=.5d0
      IF(REM1.LT.DELX)GO TO 1150
         DO 1140 K=1,NNUM
            KTI=NUMBER+K
1140        PERCNT(KTI)=PERCNT(KTI-1)+REM1
            GO TO 1200
1150        XNUM=REM/DELX+1.d0
            NNUM=INT(XNUM)
```

```
            DO 1160 K=1,NNUM
               KTI=NUMBER+K
1160           PERCNT(KTI-1)=PERCNT(KTI-2)+DELX
               NNUM=NNUM-1
1200           NUMBER=NUMBER+NNUM
      IF(PERCNT(NUMBER).GE.100.d0)PERCNT(NUMBER)=100.d0
      IF(NUMBER.GE.100.d0)GO TO 1250
      IF(PERCNT(NUMBER).GE.100.d0)GO TO 1250
         NUMBER=NUMBER+1
         PERCNT(NUMBER)=100.d0
1250  CONTINUE
C --------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C --------------------------------------------------------------
----------
1001  FORMAT(8X,'UNIT - HYDROGRAPH TERMINATED AFTER',I4,'
INTERVALS')

      RETURN
      END SUBROUTINE SUBSB
```

```
C --------------------------------------------------------------
---------------------
      SUBROUTINE
tab_hyd(Q,Area,mref,nref,ti,qp,tp,nhyd,Dstep,NA,AA)
C --------------------------------------------------------------
---------------------
C   Calculation of hydrograph by convolution (Chow, 1987) of SCS
unit
C   hydrograph and excess hyetograph
C --------------------------------------------------------------
---------------------
C      version 3.0.1, Last Modified: See Modifications below
C      WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C      Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons,
BAE (jep)
C                  University of Florida        BAE, NC State
University
C                  Gainesville, FL 32611        Raleigh, NC
27695-7625(USA)
C                  e-mail: carpena@ufl.edu
C --------------------------------------------------------------
---------------------
C   DECLARE VARIABLES
C---Inputs
C Q: runoff volume, mm
C Area: Watershed area, ha
C mref: number of unit hydrograph steps
C nref: number of excess (effective) hyetograph steps
C ti: Initial time when runoff is generated, hr
C qp: Peak flow, m3/s
C tp: Time to peak flow, hr
C nhyd: Number of timesteps in final convolution hydrograph
C Dstep:Timestep between flow calculations
C NA: Stream number
C---Other variables
C TIME1:  Time for ss of results (hrs)
C TIME2:  Time for End of results (hrs)
C cqdepth5: cumulative flow (mm/hr) of unit hydrograph
C dt5: time step (hr)
C u(5000,2):matrix holding time (t5) in column 1 and  unit
hydrograph flow (qi5, m3/s) in column
C qh(5000,3): matrix holding time (hr) in col 1, hydrograph
(m3/s) in col 2, and accumulated runoff (m3) in col 3
C unitq: volume of flow in unit hydrograph (mm)
C def: Same as dt5, time step (hr)
C A:
C qcum: cumulative runoff volume (m3/day)
```

```
C ref(5000,2): Matrix holding timestep (hr) and depth of excess
rainfall (mm)
C qpdepth: Qpeak in mm
c qhstep(5000,3): Matrix holding time (hr) in col 1, runoff
convolution hydrograph (m3/s) in col 2, and accumulated runoff
(m3) in col 3
C qh(5000,3) and qhstep(5000,3) are essentiall the same as each
other, but qhstep is set at a user defined timestep
C H(i): flow at each timestep (ft3/s)

C -----------------------------------------------------------------
---------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)
      COMMON/hydgph/u(5000,2),qh(5000,3)

COMMON/rain/rfix,rti(5000),rfi(5000),rcum(5000,2),ref(5000,2),
     c  ncum
      DIMENSION qhstep(5555,3),H(5555),AA(5555)
      !DATA H/5555*0.d0/
C-----------------------------------------------------------------
---------------------
C     FIND INODE NUMBER
C-----------------------------------------------------------------
--------------------
      DO 10 J=1,100
         IF (INT(DPRECIP(1,J)).EQ.NZ2) INODE=J
10    CONTINUE
C -----------------------------------------------------------------
---------------------
      WRITE(NUT,205)mref,nref
      cqdepth5=0.d0
      DO 40 i=1,mref
C        WRITE(*,'(2f10.4)')(u(i,j),j=1,2)
C        WRITE(NUT,'(2f10.4)')(u(i,j),j=1,2)
         cqdepth5=u(i,2)*360.d0/Area+cqdepth5 ! calculate the
cumulative volume (mm/hr) of unit hydrograph
40    CONTINUE
      dt5=u(2,1)-u(1,1) !identify the difference between time
steps
      unitq=cqdepth5*dt5 !calculate volume of unit hydrograph,
mm
      DO 50 i=1,nref !passed from hyetgh
```

```
c        WRITE(*,'(2f10.4)')(ref(i,j),j=1,2)
c        WRITE(NUT,'(2f10.4)')(ref(i,j),j=1,2)
50    CONTINUE

C ---Apply convolution of the u and ref values to obtain
hydrograph
      Def=u(2,1)-u(1,1)
      qp=0.d0
      A=0.d0
      H=0.D0
      qcum=0.d0
      DO 70 k=1,nref+mref-1
      !DO 70 k=1,nref-1
         qh(k,1)=ref(1,1)+(k-1)*Def
         qh(k,2)=0.d0
         qh(k,3)=0.d0
         DO 60 i=1,k
            qh(k,2)=qh(k,2)+ref(i,2)*u(k-i+1,2) ! llw 6.18.2022
          !print*,'qh,ref,u',qh(k,2),ref(i,2),u(k-i+1,2)
60       CONTINUE
         IF(qh(k,2).gt.qp) tp=qh(k,1) ! determine time to peak
flow by comparing qh(k,2) to qp
            qp=dmax1(qh(k,2),qp) ! set qp
        !END IF

70    CONTINUE
      qpdepth=qp*360.d0/Area
c-rmc- need to swift position of hydrograph within h so first
value correspond to no. of steps
c----- for ponding, ini (begining of hydrograph)
c      INI=INT(qh(1,1)*60.d0/5.d0)
c      print*,'INI=',INI
      !print*,'k',k
      DO 80 i=1,k-1
c -rmc- removed output of hydrographs in SI units, so only the
English units graph is shown
c------(like in unith)
c
WRITE(NUT,'(3f10.4)')(qh(i,j),j=1,2),qh(i,2)*360.d0/Area
c         H(i+INI-1)=qh(i,2)/(0.3048d0**3.d0)
         H(i)=qh(i,2)/(0.3048d0**3.d0) !converting m3/s to ft3/s
         !print*,'qh,qcum',qh(i,2),qcum
         qcum=qcum+qh(i,2)*(3600.d0*5.d0/60.d0) !qcum is in m3
80    CONTINUE
      TIME1=qh(1,1)
      TIME2=qh(i-1,1)
      qh(1,3)=0.d0
```

```
      DO 61 i=2,k-1
         qh(i,3)=qh(i-1,3)+qh(i,2)*3600*Def !MAC 04/10/12
Accumulated (m^3)
         !print*,'qh',qh(i,3)
61    CONTINUE
C ------------------------------------------------------------------
---------------------
!      WRITE(*,*)"Hydrograph"
!      WRITE(NUT,*)"Hydrograph"
!      DO 80 i=1,k-1
!         WRITE(2,'(3f10.4)')(qh(i,j),j=1,2),qh(i,2)*360.d0/Area
!MAC 04/10/12 (h, m^3/s, mm/h)
!         WRITE(*,*)(qh(i,j),j=1,3)!MAC 04/10/12 (h,m3/s,m^3)
!
WRITE(NUT,'(3f10.4)')(qh(i,j),j=1,2),qh(i,2)*360.d0/Area !MAC
04/10/12 (h, m^3/s, mm/h)
!         WRITE(NUT,*)(qh(i,j),j=1,3)!MAC 04/10/12 (h,m3/s,m^3)
!80    CONTINUE
      nhyd=k-1
      WRITE(NUT,950)ti
      WRITE(NUT,1000)qp,qpdepth,qp/(0.3048d0**3.d0)
      WRITE(NUT,1100)tp,tp*60.d0
      WRITE(NUT,1200)nhyd

!MAC 04/11/12 Interpolate the hydrograph to a time step defined
by user (Dstep(h))-
!      WRITE(*,*)"Hydrograph Aggregated"
c      WRITE(NUT,*)"Hydrograph Aggregated"
      i=1
      qhstep(i,1)=0
      DO WHILE (qhstep(i,1)<=qh(nhyd,1))
         IF (qh(1,1)<qhstep(i,1).and.qhstep(i,1)<qh(nhyd,1))
THEN
            j=1
            DO WHILE (qhstep(i,1)>qh(j,1))
              j=j+1
            END DO
            qhstep(i,3)=(qhstep(i,1)-qh(j-1,1))*(qh(j,3)-qh(j-
1,3))
            qhstep(i,3)=qhstep(i,3)/(qh(j,1)-qh(j-1,1))
            qhstep(i,3)=qh(j-1,3)+qhstep(i,3)
            qhstep(i,2)=(qhstep(i,3)-qhstep(i-1,3))/Dstep
            qhstep(i,1)=Dstep*i
c              WRITE(*,*)(qhstep(i,jj),jj=1,3)
c              WRITE(NUT,*)(qhstep(i,jj),jj=1,3)
         ELSE !This is just for the beginning, where the is no
value
```

```
            qhstep(i,3)=0
            qhstep(i,2)=0
            qhstep(i,1)=Dstep*i
         END IF
c        WRITE(*,*)(qhstep(i,jj),jj=1,3)
c            WRITE(NUT,*)(qhstep(i,jj),jj=1,3)
         i=i+1
         qhstep(i,1)=Dstep*i
      END DO
! MAC 04/11/12 just to consider the last pulse
      IF (qhstep(i-1,1)<qh(nhyd,1).and.qhstep(i,1)>=qh(nhyd,1))
THEN
        qhstep(i,3)=qh(nhyd,3)
        qhstep(i,2)=(qhstep(i,3)-qhstep(i-1,3))/Dstep
      END IF
c     WRITE(*,*)(qhstep(i,jj),jj=1,3)
c     WRITE(NUT,*)(qhstep(i,jj),jj=1,3)
!  Save in SS qh !MAC 04/10/12
!    DO 62 j=1,i
         !SS storages hydrograph in CFS
C ----------------------------------------------------------------
---------------------
C   CONVERSION
C ----------------------------------------------------------------
---------------------
!    SS(j,NA)=SS(j,NA)+(qhstep(j,2)/(0.3048d0**3)) !Unit
conversion from m^3/s to CFS
!    Hydro(j,2)=(qhstep(j,2)/(0.3048d0**3))
!    Hydro(j,2)=(qhstep(j,2)) !Metric Units
!    Hydro(j,1)=qhstep(j,1)
!        write(*,*) Hydro(j,1),Hydro(j,2)
C        WRITE(NUT,*)  Hydro(j,1),Hydro(j,2)
!62  CONTINUE
!    IF (SS(5555,NA)<i) SS(5555,NA)=i
!     CALL MREAD(NA,AA)
!     AA(5555)
!  AA(5555)=Numero de posiciones
C ----------------------------------------------------------------
-------------------------
C Pass hydrograph to STREAM flow MATRIX SS
C ----------------------------------------------------------------
-------------------------
      H(5555)=nref+mref-1
      INTERV=nref+mref-1
      IF(INTERV.GT.440) INTERV=440
      UNIT=5.d0
      ITIME1=int(60.d0*TIME1/UNIT)
```

```
      ITIME2=int(60.d0*TIME2/UNIT)

      !print*,'itime1,itime2',itime1,itime2
      !CALL MREAD(NA,AA) !LLW 5.7.2023
      NUMX=INT(UNIT/5.d0+.01d0)
C      DO 750 I=1,INTERV
C         AA(I)=AA(I)+H(I)
C750   CONTINUE
c--- store hydrograph in aa() matrix at correct time step
      icount=1
      if (itime1.eq.0.d0) then
          itime1=1
          end if
      do 750 i=itime1,itime2
         aa(i)=aa(i)+h(icount) !llw 5.7.2023
         !h(icount)=h(icount)+stail(i,inode) !adding stail to
runoff generated at this node
         icount=icount+1
750   continue
c--- add existing runoff (AA(I)) to tail from previous day
(stail(i,inode))
      DO 770 I=1,5555
         AA(I)=AA(I)+STAIL(I,INODE)
770   CONTINUE
      AA(5555)=INTERV*NUMX
      CALL MWRITE(NA,AA)
C ------------------------------------------------------------------
-------------------------
C Print hydrograph (units in CFS and AF)
C ------------------------------------------------------------------
-------------------------
      KTYPE=0
      XMAX=qp/(0.3048d0**3.d0)
      SUM=qcum/1233.48d0 !converts qcum from m3/d to ac-ft/day
      CALL OASB(KTYPE,H,INTERV,XMAX,UNIT,SUM,TIME1,TIME2)
C ------------------------------------------------------------------
-------------------------
C Pass tail along for next day
C ------------------------------------------------------------------
-------------------------
      DO 780 I=1,5555-1
         STAIL(I,INODE)=0.D0
         IF (I.GT.288) THEN
             STAIL(I-288,INODE) = AA(I)
         END IF
780   CONTINUE
      if (AA(5555).ge.288) then
```

```
            stail(5555,INODE)=AA(5555)-288.d0
        else
            stail(5555,INODE)=0.d0
        end if
C ----------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C ----------------------------------------------------------------
----------
c-rmc- change to stop displaying the hydrograph in SI units
(like in unith)
c205   FORMAT(4X,'Number of unit hydrograph steps (n)  = ',i5,/,
c     1 4X,'Number of excess hyetograph steps (m) =',i5,/,/,
c     2 4X,'b) Final Hydrograph (CONVOLUTION):',/,
c     3 4X,'Time(h)   q(m3/s)   q(mm/h)',/,3X,30('-'))
205   FORMAT(4X,'Number of unit hydrograph steps (n)  = ',i5,/,
     1 4X,'Number of excess hyetograph steps (m) =',i5,/,
     2 4X,30('-'),/,4X,'b) Final Hydrograph (CONVOLUTION):')
950   FORMAT(4X,'Time to Ponding =',f8.3,' hr')
1000  FORMAT(4X,'Peak flow    =',f15.3,' m3/s = ',f15.4,' mm/h =
',
     & f15.4,' cfs')
1100  FORMAT(4X,'Time to peak =',f8.2,' h    = ',f8.2,' min')
1200  FORMAT(4X,'Number of final hydrograph steps (nhyd) =
',i5,/)

      RETURN
      END SUBROUTINE tab_hyd
```

```
      subroutine thetafao(CINF,isoil,UFC,UWP,Zr,pfrac,Hm,
     & THETA,ETA,dperc,inode,dtheta1,FC,WP,P,BFsm,DIRREFF)
C-----------------------------------------------------------------
c      version 0.7, Last Modified 11/14/2022
C      WRITTEN FOR: EU FOCUS PRZM/VFSMOD tool
C      Written by: R. Munoz-Carpena (rmc)
C                  University of Florida, carpena@ufl.edu
C-----------------------------------------------------------------
C Program to calculate soil moisture content between runoff
events.
C This is a necessary step for continuous simulation of the
PRZM/VFSMOD
C EU FOCUS tool. The result of the calculation will produce the
initial
C moisture content of the soil (OI) for the next VFSMOD run in
the time
C series. It follows FAO-56 adjusted ET calculations (Allen et
al.,1998)
C based on Dr. M. Qußemada (U. Politecnica Madrid) spreadsheet
calculations
C and good results in the comparison with field measured soil
moisture.
c-----------------------------------------------------------------
C   Input parameters
c   isoil (soilty) (USDA, S:Sand;L:Loam;s:Silt;C:Clay):-
1:user,1:S,2:LS,3:SL,4:L,5:sL,6:s,7:sCL,8:sC,9:C
c   OI(m3/m3): top soil initial water content (same as in VFSMOD
*.iso file)
c   FC(m3/m3): top soil fied capacity water content (read
internally or provided by user when isoil=-1)
c   WP(m3/m3): top soil wilting point water content (read
internally or provided by user when isoil=-1)
c   Zr(m): maximum grass root zone depth (typical values (0.5-
1.5 m)
c   pfrac[-]: fraction of easily estractable water (typical 0.6
for Bermuda grass)
c   Hm(m): height of vegetation (from VFSMOD *.igr file,
H(cm)/100)
!c   iFH(optional): input MET file formating flag where,
!C     iFM= 0 (or not present), 8 columns,last two columns are
Tmin,Tmax
!C     iFM= 1, 7 columns, last column is HRmean
!C     iFM= 2, 9 columns, last 3 columns are Tmin,Tmax, Kcmid
for a crop other than grass
!C     iFM= 3, 8 columns, last 2 columns are HRmin, Kcmid for a
crop other than grass
c   REW(mm): readily extractable water (soil dependent)
```

```
c   soil(isoil,1): FC, top soil field capacity (m3/m3)
c   soil(isoil,2): WP, top soil wilting point (m3/m3)
c   soil(isoil,1): top soil REW(mm) (see above)
c   TAW(mm): total available water
C-----------------------------------------------------------------
c    Compiling for Win32 and Unix environments:
c       1. The i/o for these operating systems is different.
c       2. Change the comments in the finput.f program to
reflect
c          your operating system.   3/9/2012
c-----------------------------------------------------------------
c    CHANGES
c    v0.7, 11/14/2022. Added AET (mm) to last column of modified
.MET output file and fix field
c      length for date that was short of 1 character on the
output .MET file
c    v0.6, 12/19/2019. Changed iRH into iFM (input formatting
MET)
c      to modify the last columns of the MET file with user
cKmid
c-----------------------------------------------------------------

      implicit double precision (a-h, o-z)
      common /gampar/ vsatk,sav,wcsat,wcini,bm,deltim,stmax
      dimension soil(21,2),DIRREFF(5555,100)

c-------------------------------------------
c  Select soil parameters
c-------------------------------------------
      data(soil(1,J),J=1,2)/0.36d0,0.22d0/
      data(soil(2,J),J=1,2)/0.36d0,0.23d0/
      data(soil(3,J),J=1,2)/0.36d0,0.25d0/
      data(soil(4,J),J=1,2)/0.335d0,0.205d0/
      data(soil(5,J),J=1,2)/0.36d0,0.22d0/
      data(soil(6,J),J=1,2)/0.27d0,0.17d0/
      data(soil(7,J),J=1,2)/0.32d0,0.17d0/
      data(soil(8,J),J=1,2)/0.29d0,0.15d0/
      data(soil(9,J),J=1,2)/0.25d0,0.12d0/
      data(soil(10,J),J=1,2)/0.23d0,0.11d0/
      data(soil(11,J),J=1,2)/0.23d0,0.11d0/
      data(soil(12,J),J=1,2)/0.23d0,0.11d0/
      data(soil(13,J),J=1,2)/0.23d0,0.11d0/
      data(soil(14,J),J=1,2)/0.15d0,0.065d0/
      data(soil(15,J),J=1,2)/0.15d0,0.065d0/
      data(soil(16,J),J=1,2)/0.15d0,0.065d0/
      data(soil(17,J),J=1,2)/0.15d0,0.065d0/
      data(soil(18,J),J=1,2)/0.12d0,0.045d0/
```

```
      data(soil(19,J),J=1,2)/0.12d0,0.045d0/
      data(soil(20,J),J=1,2)/0.12d0,0.045d0/
      data(soil(21,J),J=1,2)/0.12d0,0.045d0/

          OI=wcini
      !if(isoil.eq.-1) then
      !      !OI=uOI
            FC=uFC
            WP=uWP
      !   elseif(isoil.le.21) then
      !      FC=soil(isoil,1)
      !      WP=soil(isoil,2)
      !      !OI=FC
      !   else
      !          write(*,*)'ERROR: wrong soil type selection (-
1,21)'
      !          STOP
      !endif
c---------------------------
c Calculate runoff volume by SCS method
c---------------------------
      call theta10(OI,Zr,pfrac,Hm,cKmid,WP,FC,theta,ETa,
     & inode,CINF,dperc,P,BFsm,DIRREFF)

c---Pass change in soil moisture to parent routine
      dtheta1=theta-OI

c---------------------------
c Output results
c---------------------------
      call
TFresults(isoil,OI,FC,WP,Zr,pfrac,Hm,TAW,cKmid,CINF,theta,
     & dtheta1)

      !print*,'oi,theta,dtheta1',OI,theta,dtheta1

      end subroutine



      subroutine theta10(OI,Zr,pfrac,Hm,cKmid,WP,FC,theta,ETa,
     & inode,CINF,dperc,P,BFsm,DIRREFF)
C-------------------------------------------------------------------
C Soil moisture calculation, FAO (1998)
C MET file: the last columns can be changed to be read from the
C   modified EUFOCUS MET files (last 2 columns Tmax,Tmin),
and/or
```

```
C    provide crop coefficient values (kcmid) for plants other
than
C    than grass using the different values of iFM:
C  iFM= 0 (or not present), 8 columns,last two columns are
Tmin,Tmax
C  iFM= 1, 7 columns, last column is HRmean
C  iFM= 2, 9 columns, last 3 columns are Tmin,Tmax, Kcmid for a
crop other than grass
C  iFM= 3, 8 columns, last 2 columns are HRmin, Kcmid for a crop
other than grass
C-----------------------------------------------------------------
      implicit double precision (a-h, o-z)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE

common/CINPUT/DETO(5555,100),DBF(5555,100),DTAVG(5555,100),
     &
DTMAX(5555,100),DTMIN(5555,100),DWS2(5555,100),DSORAD(5555,100),
     &
DCKM(5555,100),DAB(5555,100),DIRR(5555,100),DSNO(5555,100)
      !character*7 datein
      character*100 dum
      DIMENSION DIRREFF(5555,100)

c-----Initial calculations before the data processing loop
c-----root zone and top layer initial depletion
      Dro=(FC-OI)*Zr*1000.d0
      TAW=(FC-WP)*Zr*1000.d0
      FCmm=fc*Zr*1000.d0
      dperc=0.d0
      BFsm=0.d0
c-----Start calculation loop for original .MET file and save
file with additional column in
c-----output directory. The scratch file is used to read PRZM
date fix format (first column 'A7')
c-----while the other inputs are are read as free format.
c-----read inputs from matrices input files for the given day.
JDAY+1 = current day because each matrix has node headers

      EToMM = DETO(JDAY+1,INODE)
      RRIGATION = DIRREFF(JDAY,INODE)
      if (p.gt.3.d0.AND.RRIGATION.EQ.0.D0) then
          etomm=0.d0
      end if
      TEMP = DTAVG(JDAY+1,INODE)
      U2 = DWS2(JDAY+1,INODE)
      RAD = DSORAD(JDAY+1,INODE)
```

```
      TEMPMAX = DTMAX(JDAY+1,INODE)
      TEMPMIN = DTMIN(JDAY+1,INODE)
      CKMID = DCKM(JDAY+1,INODE)
      !print*,'rad,u2,temp',rad,u2,temp

!read(25,*)prec,ETo,temp,u2,rad,tempmax,tempmin,cKmid
              RHm=RHmin(tempmax,tempmin)
!           CASE (3)
!              read(25,*)prec,ETo,temp,u2,rad,RHm,cKmid
!           CASE DEFAULT
!              read(25,*)prec,ETo,temp,u2,rad,tempmax,tempmin
!cdebug
write(*,'(i6,8f8.2)')i,prec,ETo,temp,u2,rad,tempmax,tempmin
!              RHm=RHmin(tempmax,tempmin)
!        END SELECT
!        close(25)
c------change units to mm/day (P,ETo) and m/s (u2) from default
in MET file ------
c        precmm=prec*10.d0
        precmm=CINF !added by LW 11.22.2022
       ! print*,'precmm,etomm',precmm,etomm
        !ETomm=ETo*10.d0
        u2m=u2/100.d0
          if(u2m.lt.0.d0)u2m=0.d0
c-----Calculate adjusted crop coefficient (cKb) based on daily
wind and relative temperature
        cKbadj=(0.04d0*(u2m-2.d0)-0.004d0*(RHm-
45.d0))*(Hm/3.d0)**0.3d0
        !print*,'ckbadj',ckbadj
        cKb=cKmid+cKbadj
        ETc=cKb*ETomm
c-----Calculate root zone layer depletion (Dr) from the daily
soil water balance
        dperc= precmm-ETc-Dro !deep percolation = precipitation
- ETc - initial root zone depletion
        if(dperc.lt.0.d0) dperc=0.d0
        Dr=Dro-precmm+ETc+dperc !initial root zone depletion -
precipitation + ETc + precip - ETc - Dr
        if(Dro.lt.0.d0) Dr=0.d0
        if(Dro.gt.TAW)  Dr=TAW
c-----Calculate readily available water (RAW)
        pefftf=pfrac+0.04d0*(5.d0-ETc)
        RAW=TAW*pefftf
        if(Dr.gt.RAW)then
            cKs=(TAW-Dr)/(TAW-RAW)
          else
            cKs=1.d0
```

```
          endif
c-----Calculate the actual ET (ETa) and recalculate Dr and dperc
based on ETa
        ETa=cKb*cKs*ETomm !
        Dr=Dro-precmm+ETa+dperc
        dperc=Dr-Dro+precmm-ETa !!!LW testing 7.3.2023
         !Eta=ETomm
        !print*,'ckb,cks,etomm',ckb,cks,etomm
        !print*,'eta',eta
c-----Calculate the soil moisture (depth Zr)
        thetamm=1000.d0*FC*Zr-Dr
        theta=thetamm/(Zr*1000.d0)
        If (theta.gt.FC) then
            BFsm=theta-FC
            theta=FC
            print*,'jday, node', jday,inode
        end if

c-----Write the new theta10 and ETa columns to the output/*.met
file
cdebug
write(*,200)i,precmm,ETomm,u2m,RHm,cKb,ETc,pfrac,peff,
cdebug     &     RAW,TAW,Dr,dperc,cKs,ETa,theta
C        if(iFM.eq.1.or.iFM.eq.3) then
c           write(4,102)datein,prec,ETo,temp,u2,rad,RHm,theta
!
write(4,102)datein,prec,ETo,temp,u2,rad,RHm,theta,ETa/10.d0
!         else
!c
write(4,100)datein,prec,ETo,temp,u2,rad,tempmax,tempmin,
!!c     &       theta

write(nut,100)precmm,ETоMM,u2,rad,tempmax,tempmin,RHm,
     &       theta,ETa
!        endif
!10    continue

100   FORMAT(/,4x,'Infiltration              =  ',f10.2,' mm',
     C  /,4x,'Reference ET             =  ',f10.2,' mm',
     C  /,4x,'Wind Speed               =  ',f10.4,' cm/s',
     C  /,4x,'Solar Radiation          =  ',f10.4,' units',
     C  /,4x,'Average max temperature  =  ',f10.4,' C',
     C  /,4x,'Average min temperature  =  ',f10.4,' C',
     C  /,4x,'Relative humidity        =  ',f10.2,' %',
     C  /,4x,'Soil Moisture            =  ',f10.4,' m3/m3',
     C  /,4x,'Actual ET                =  ',f10.4,' mm')
101   format(a7,a100)
```

```
102   format(a7,2f10.2,f10.1,f10.0,f10.1,f10.2,f10.3,f10.3)
120   FORMAT(4X,'Solution does not converge in max iterations')
cdebug 200   format(i6,20f10.3)


30    return
      end subroutine


      function RHmin(tempmax,tempmin)
c----------------------------------------------------------------
c--- Estimate missing RHmin based on assumption Tmin<>Tdew
c----------------------------------------------------------------
      implicit double precision (a-h, o-z)

      a1= 17.625d0
      b1= 243.04d0
      c1= 0.61121d0
      ea= c1*dexp(a1*tempmin/(tempmin+b1))
      es= c1*dexp(a1*tempmax/(tempmax+b1))
      RHmin=100.d00*ea/es
      if(RHmin.gt.100.d0) RHmin=100.d0

      return
      end


      subroutine
TFresults(isoil,OI,FC,WP,Zr,pfrac,Hm,TAW,cKmid,CINF,
     & theta,dtheta1)
C----------------------------------------------------------------
C      Output summary of hydrology results nicely
C----------------------------------------------------------------
      implicit double precision (a-h, o-z)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      character*13 stype(21)
      data stype/'Clay','Silty clay','Sandy clay','Silty clay
loam',
     C  'Clay loam','Sandy clay loam','Silt','Silt loam','Loam',
     C  'Very fine sandy loam','Fine sandy loam','Sandy loam',
     C  'Coarse sandy loam','Loamy very fine sand','Loamy fine
sand',
     C  'Loamy sand','Loamy coarse sand','Very fine sand',
     C  'Fine sand','Sand','Coarse sand'/

      TAW=(FC-WP)*Zr*1000.d0
```

```
      write(nut,*)' '
      write(nut,*)'TOP SOIL MOISTURE CALCULATION FAO (1998)
METHOD'
      write(nut,*)' '
      write(nut,*)'INPUTS'
      write(nut,*)'------'
      if(isoil.eq.-1) then
               write(nut,200)'User'
          else
               write(nut,200)stype(isoil)
      endif
      write(nut,250)FC
      write(nut,300)WP
      write(nut,325)OI
      write(nut,450)Zr
      write(nut,500)pfrac
      write(nut,600)TAW
!      if (iFM.eq.2.or.iFM.eq.3) then
          write(nut,650)cKmid
      !  else
          !write(nut,650)cKmid
  !    endif
      write(nut,700)Hm
      write(nut,750)CINF
      write(nut,760)theta
      write(nut,770)dtheta1
      !SELECT CASE (iFM)
         !CASE (1)
         !    write(3,800)iFM,'last column RHmean            '
         !CASE (2)
      !       write(nut,800)iFM,'last 3 columns Tmax,Tmin,cKmid'
      !   CASE (3)
      !       write(3,800)iFM,'last 2 columns RHmean,cKmid   '
      !   CASE DEFAULT
      !       write(3,800)iFM,'last 2 columns Tmax,Tmin      '
      !END SELECT
      write(nut,*)' '

200   format('Soil type',39x,'=',4x,a13)
250   format('Top soil field capacity, FC(m3/m3)',14x,'=',f9.3)
300   format('Top soil wilting point, WP(m3/m3)',15x,'=',f9.3)
325   format('Top soil initial water content,
OI(m3/m3)',7x,'=',f9.3)
450   format('Maximum grass root zone depth,
Zr(m)',12x,'=',f8.2)
500   format('Fraction of easily extractable
water,pfrac',6x,'=',f8.2)
```

```
600   format('Total available water, TAW(mm)',18x,'=',f8.2)
650   format('Mid season crop coeff., Kcmid',19x,'='f8.2)
655   format('Mid season crop coeff., Kcmid',19x,
     & '=    variable (.MET file)')
700   format('Vegetation height, H(m)',25x,'=',f8.2)
750   format('Infiltrated water volume/precipitation,
mm',6x,'=',f8.2)
760   format('Final soil moisture, theta, m3/m3',15x,'=',f9.3)
770   format('Change in soil moisture, dtheta,
m3/m3',10x,'=',f9.3)
!800   format('Input format option for MET file, iFM
=',i2,7x,'=',a34)

      return
      end subroutine
```

```
C  PROGRAM 19 - Based on Hromadka book 232 pag
C  ---------------------------------------------------------------
-------------
      SUBROUTINE TRAPV(Q,B,Z,E1,E2,XL,RN,V,DN)
C ----------------------------------------------------------------
-------------
C DECLARE VARIABLES
C ----------------------------------------------------------------
-------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
C S=CHANNEL SLOPE, FRACTION
C B= CHANNEL WIDTH, ft
C Z:      Channel “Z” factor – Ratio of Horizontal/vertical. [0
- 100]              C
C E1:     Upstream elevation (ft) [-3  to 3000]
                         C
C E2:     Downstream elevation (ft) [-60 to 3000]
                         C
C XL:     Channel length  - the length of the  longest
watercourse (ft)              C
C XN(/RN):     Basin Factor (Manning's Friction Factor) [0.008 -
0.999]         C
C ----------------------------------------------------------------
-------------
      S=(E1-E2)/XL
      IF(B.LE.0.d0)B=.0001d0
      VMAX=1000.d0
      YMAX=VMAX
      YMIN=0.d0
      DO 440 I=1,17
         DN=.5d0*(YMIN+YMAX)
         F=1.d0-Q*RN*(B+2.d0*DN*DSQRT(Z*Z+1.d0))**.555567d0/
     C    (1.486d0*((B+Z*DN)*DN)**1.55557d0*DSQRT(S))
         IF(F)420,450,430
420      YMIN=DN
         GO TO 440
430      YMAX=DN
440   CONTINUE
450   TW=B+2.d0*Z*DN
      AREA=.5d0*(B+TW)*DN
      V=Q/AREA
C ----------------------------------------------------------------
-------------
      RETURN
      END SUBROUTINE TRAPV
```

```
C ------------------------------------------------------------------
------------------
C  Program:   SCS METHOD
C ------------------------------------------------------------------
------------------
 !     SUBROUTINE uhcn (m,n,m1,n1,mn1,mn2) !ARGU = (NUT,NDAT)
(2.1.18)
      SUBROUTINE uhcn (P,DAYRO,DAYDN,DSM1,DSM2,DRECH,DBASEF,
     &            DAYMO,DTHETA,DETA,SISTORE,BFloss,DSED,PEFF,
     &
DTHETA2,dstorvol,dmaxstor,drloss,dSIWater1,dminstor,
     &            DBF,DIRREFF)
C ------------------------------------------------------------------
------------------
c  version 3.0.1, Last ModIFied: See ModIFications below
C  WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C  Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons, BAE
(jep)
C     University of Florida                 BAE, NC State
University
C     Gainesville, FL 32611                 Raleigh, NC 27695-
7625(USA)
C     e-mail: carpena@ufl.edu
C ------------------------------------------------------------------
------------------
C Program to create input files for VFSmod, based on NRCS-TR55
and Haan
C et al, 1996, with additional work DOne on coefficients for
unit peak
C flow calculation.
C ------------------------------------------------------------------
------------------
C Subroutines and functions:

C-------------------------------------------------------------------
-----------------
c    Date      Modification
Initials
c   -------    ------------------------------              ---
---
c   2/17/99    Check for 0.1<Ia/P<0.5
rmc
c   2/18/99    Added hyetograph output for 6 h storm
jep
c   2/18/99    ModIFy File Inputs for Erosion
jep
```

```
c   2/20/99    Roughed in MUSLE
jep
c   3/01/99    Checked erosion parameters and units
rmc
c   3/02/99    Additional work on Musle - units close
jep
c   3/03/99    Added hyetographs for storm types I & IA
rmc
c   3/05/99    output irs file for VFSMOD
jep
c   3/06/99    Input/Output files as in VFSMOD
jep
c   3/10/99    Checked Input/Output files as in VFSMOD
rmc
c   3/10/99    Cleanup - created hydrograph.f for
c              hydrograph subroutines, created io.f for
c              input and output related processing
jep
c   3/28/99    Erosion part: fixes in I30 calculation
c              after Chow and checked for consistency in
c              units, clean up; Hydro: added delay time
rmc
c   8/27/99    Added option to select dIFferent methods
c              for applying MUSLE, default is Foster,
c              2=Williams, 3=GLEAMS
c   10/01/99   Fixed array so that storm duration (D)
c              can now be up to 24h
rmc
c   10/26/99   implemented the project file concept as in vfsm
jep
c    3/09/00   Version changed to 0.9, general program cleanup
rmc
c   16/06/00   Version changed to 1.0, erosion output organized
rmc
c   16/03/02   Version changed to 1.06 to couple with VFSMOD,
c              author affiliation changed
rmc
c    4/18/03   Fixed K - computed IF we enter -1, other use
jep
c              entered value, also fixed dp output format
c    4/19/03   dp now being read in
jep
c    4/20/03   Runoff calculation for low CN revised
rmc
c    5/01/03   Added chacked for small runoff case to switch
rmc
c              to Williams sediment calculation that includes
```

```
c              runoff.
c   11/10/03   Reordered Erosion ieroty 1=Williams, 2=Gleams
c                3=Foster to coincide with changes in Shell
jep
c   11/13/03   Fixed coef. on Type Ia - did not add new
c                hyet curves
c   01/10/05   Added changes suggested by U. of Guelph group
rmc
c              v2.4.1
c   09/15/11   Rewritten hydrograph calculation using
convolution
c              of excess rain steps, v3.0.0
rmc
c   02/15/12   Added user table for 24-h hyetograph, v3.0.1
rmc
C --------------------------------------------------------------
-------------------
c Compiling for Win32 and Unix environments:
c       1. The i/o for these operating systems is dIFferent.
c       2. Change the comments in the finput.f program to
reflect
c          your operating system.   3/9/00
C --------------------------------------------------------------
-------------------
C
Inputs:(NA,CN,Area,jstype,D,pL,Y,ek,cfact,pfact,soilty,ieroty,dp
,om,JCOUNT)
c       NA, number of stream being modeled
c       CN, dimensionless curve number
c       Area, contributing watershed area, ha
c       jstype,  SCS storm type (1=I, 2=IA, 3=II, 4= III, or 5=
'user')
c       D, storm duration, h
c       pL, Longest flow path, ft
c       y, watershed slope, ft/ft
c       ek, soil erodibility, set as -1 to default to correspond
to soil type
c       cfact, dimensionless cover management factor, "C factor"
c       pfact, dimensionless support practices factor, "P
factor:
c       soilty, soil texture, see texture/type correspondance in
musle.f
c       ieroty, Select the method to estimate storm erosion R
factor in MUSLE, not present or =1 selects Foster's Method,=2
selects Williams method, and =3 selects the CREAMS/GLEAMS method
c       dp, particle size, d50 in cm; if set as -1 then it
depends on soil type
```

```
c       om, % soil organic matter, read IF ek <0; set as 2.0 if
not set
c       jcount, current day in simulation run
c       P, precipitation, mm

c COMMON/hydgph:
c       rot(208), runoff time (units)
c       roq(208), runoff rate (m3/s)
c       u(208,2), unit hydrograph
c COMMON/rain/:
c       rfix, maximum rain intensity (mm/h)
c       rti(200), rainfall time (hrs)
c       rfi(200), rainfall intensity (mm/h)
c       rcum(100,2), cumm rainfall (mm)
c       ref(100), excess rainfall intensity (mm/h)
c       ncum: number of steps IF user hyetograph is read
c other:
c       nref  = number of excess hyetograph steps
c       mref  = number of unit hydrograph steps
c       nhyet = number of hyetograph steps
c       vol(m3), volro (mm) = runoff volume
C --------------------------------------------------------------
-----------------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
-----------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/BLK1/SS(5555,15),SS1(5555,15),DPRECIP(5555,100),
     & SNODE(5555,100),STAIL(5555,100)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      DIMENSION
dstorvol(100),dmaxstor(100),dminstor(100),drloss(100),
     & dSIWater1(100)
      !CHARACTER*20 soilty
      CHARACTER*75 LISFIL(19)
      DIMENSION DAYRO(5555,100),DAYDN(5555,100),PEFF(5555,100),
     &
DTHETA(5555,100),DETA(5555,100),SISTORE(100),DBASEF(5555,100),
     &
DSED(5555,100),DRECH(5555,100),DSM1(5555,100),DSM2(5555,100),
     & DAYMO(5555,100),DTHETA2(5555,100),DBF(5555,100)
      DIMENSION AA(5555),SUMQO24(288),DIRREFF(5555,100)
      DO 50 K=1,100
        IF (INT(DPRECIP(1,K)).EQ.NZ2) THEN
            inode=K
```

```
        END IF
      AA=0.D0
      SUMQO24=0.D0
50    CONTINUE
C ----------------------------------------------------------------
------------------------
C  Get inputs and open files
C ----------------------------------------------------------------
------------------------
      CALL getinp(NA,CN,Area,jstype,D,pL,Y,ek,cfact,pfact,isoil,
     C            ieroty,dp,om,uFC,uWP,ZR,PFRAC,HM,
     C            dtheta,soilpt,Zstore,inodeloc,arec,brec)
C ---Calculate volume of effective precipitation for Water
balance
      PEFF(JDAY,INODE)=P*AREA*10.D0

      !PRINT*,'PUHCN',P

C ---Correct Curve Number based on Antecendent Moisture
Condition
      AMC=0.D0
      IF (JDAY.EQ.1) THEN
      AMC = 36.D0
      !PRINT*,'CN',CN
      ELSE IF (JDAY.LT.6) THEN
          DO icount = 1, JDAY-1
              AMC = AMC + (PEFF(ICOUNT,INODE))/(AREA*10.D0)
          END DO
      ELSE
          AMC=(PEFF(JDAY-1,INODE)+PEFF(JDAY-2,INODE)+PEFF(JDAY-
3,INODE)
     &    +PEFF(JDAY-4,INODE)+PEFF(JDAY-5,INODE))/(AREA*10.D0)
      END IF
      IF (AMC.LT.36.D0) THEN
          CN = 4.2D0*CN/(10.D0-CN*0.058D0)
      ELSE IF (AMC.GT.53.D0) THEN
          CN = 23.D0*CN/(10+CN*0.13D0)
      END IF
      !print*,'amc',amc
      !PRINT*,'CN2',CN
C ----------------------------------------------------------------
------------------------
C  Calculate runoff volume by SCS method
C ----------------------------------------------------------------
------------------------
      CALL runoff(P,CN,xIa,Q) !OK
!      print*,'Pasado runoff'
```

```
      volro=Q
      !print*,'q',q
      IF (Q.GT.0.D0) THEN
C --------------------------------------------------------------
-----------------------
C  Calculate concentration time by SCS method
C --------------------------------------------------------------
-----------------------
         CALL calctc(pL,CN,Y,tc)
         !Dstep=0.24d0*tc !MAC 04/10/12
          Dstep=5.d0/60.d0
C --------------------------------------------------------------
-----------------------
C  Calculate peak flow and time by SCS-TR55 method
C --------------------------------------------------------------
-----------------------
          CALL q_peak(Area,Q,xIa,P,tc,jstype,qp,tp) !OK
          !print*,'qp',qp
C --------------------------------------------------------------
-----------------------
C  Output hydrology results
C --------------------------------------------------------------
-----------------------
          CALL results(P,CN,Q,Area,tc,xIa,jstype,D,pL,Y,
     C             ieroty)
C --------------------------------------------------------------
-----------------------
c  Calculate SCS-unit hydrograph
C --------------------------------------------------------------
-----------------------
          CALL unit_hyd(Q,Area,qp,tp,D,tc,mref)
C --------------------------------------------------------------
-----------------------
c  Calculate storm hyetograph from SCS storm type
C --------------------------------------------------------------
-----------------------
          CALL hyetgh(jstype,P,D,xIa,
     &       ti,nref,a1,b1,bigE,raimax30,ndtime)
          !print*,'ti',ti
C --------------------------------------------------------------
-----------------------
c  Calculate storm hydrograph
C --------------------------------------------------------------
-----------------------
         CALL
tab_hyd(Q,Area,mref,nref,ti,qp,tp,nhyd,Dstep,NA,AA)
```

```
C ----------------------------------------------------------------
------------------------
C DO the modified usle to get erosion stuff
C ----------------------------------------------------------------
------------------------
      CALL musle(er,er1,erCoolm,ek,Y,pl,cfact,pfact,Area,Q,tc,P,
     C
D,isoil,dp,sconc,sconc1,sconc2,om,a1,b1,bigE,raimax30,qp,
     C      ieroty,sconc3)

C ----------------------------------------------------------------
------------------------
C iF NO NEW RUNOFF OCCURS - add tail from previous day
C ----------------------------------------------------------------
------------------------
      ELSE
          CALL results(P,CN,Q,Area,tc,xIa,jstype,D,pL,Y,
     C             ieroty)
          WRITE(NUT,1000)
          DO 770 I=1,5555
              AA(I)=AA(I)+STAIL(I,INODE)
770       CONTINUE
          CALL MWRITE(NA,AA)
      DO 780 I=1,5555-1
         STAIL(I,INODE)=0.D0
         IF (I.GT.288) THEN
             STAIL(I-288,INODE) = AA(I)
         END IF
780   CONTINUE
      if (AA(5555).ge.288) then
          stail(5555,INODE)=AA(5555)-288.d0
      else
          stail(5555,INODE)=0.d0
      end if
      END IF
c-----------------------------------------------------------------
------------------------
C Use ThetaFAO to calculate initial soil moisture for next day
c ----------------------------------------------------------------
------------------------
      IF (P.GT.0.0D0) THEN
          CINF=P-Q
      ELSE
          CINF=0.0D0
      END IF
      CALL thetafao(CINF,isoil,UFC,UWP,Zr,pfrac,Hm,
     & THETA,ETA,DPerc,inode,dtheta1,FC,WP,P,BFsm,DIRREFF)
```

```
C ---------------------------------------------------------------
-----------------------
C  Calculate deep percolation seepage/fractional redistribution
C ---------------------------------------------------------------
-----------------------
    !DBASEF(JDAY,INODE)=DBASEF(JDAY,INODE)-DBASEF(JDAY,JNODE)
       CALL DPSEEP(ISOIL,SISTORE,DPerc,INODE,BFloss,NA,AREA,
     &
soilpt,Zstore,RECHARGE,DSIwater,FC,WP,wcini,SWC2,DBASEF,DRECH,
     &
DSM2,dstorvol,dmaxstor,drloss,dSIWater1,dminstor,DBF,DTHETA2)

C ---------------------------------------------------------------
-----------------------
C  Calculate 24 hour flows for first day
C ---------------------------------------------------------------
-----------------------
      SUMQO24(1)=0.5D0*AA(1)*5.D0*60.D0*0.0283168D0
      DO 11 I=2,288
          SUMQO24(I)=SUMQO24(I-1)+0.5d0*(AA(I-1)+
     &        AA(I))*5.d0*60.d0*0.0283168d0 !Calculating the sum
by taking average of timesteps, multiplying over time step to
get volume, and converting to cubic meters
11    CONTINUE


C ---------------------------------------------------------------
-----------------------
C  Calculate deep percolation seepage/fractional redistribution
C ---------------------------------------------------------------
-----------------------
!       CALL DPSEEP(ISOIL,SISTORE,DPerc,INODE,BFloss,NA,AREA,
!      &
soilpt,Zstore,RECHARGE,DSIwater,FC,WP,wcini,dailyq,arec,brec)
!      !
C ---------------------------------------------------------------
-----------------------
C  Write runoff to STORAGE file
C ---------------------------------------------------------------
-----------------------
      !DAYRO(JDAY,INODE) = Q*AREA*10.D0
      DAYRO(JDAY,inode)=SUMQO24(288)
      DAYDN(JDAY,INODE)=CINF*Area*10.d0
      DAYMO(JDAY,INODE)=Q*AREA*10.D0-SUMQO24(288)!Accounts for
any runoff that has not entered the stream within 24 hours
      DTHETA(JDAY,INODE)=THETA
```

```
!DBASEF(JDAY,INODE)=BFsm*Area*Zr*10000.d0+BFLOSS+DBASEF(JDAY,INO
DE)
      !DTHETA2(JDAY,INODE)=SWC2
      DETA(JDAY,INODE)=ETA*Area*10.d0
      !DRECH(JDAY,INODE)=RECHARGE
      !PRINT*,'RECHARGE UHNC', DRECH(JDAY,INODE)
      DSM1(JDAY,INODE)=dtheta1*Area*Zr*10000.d0
      !DSM2(JDAY,INODE)=DSIwater

C ---------------------------------------------------------------
------------------------
C  CALCULATE BASEFLOW RECESSION
!C --------------------------------------------------------------
-------------------------
!      IF (INODELOC.EQ.1) THEN
!          IF (DAYRO(JDAY,INODE).GT.0.D0) THEN
!           dailyq=SUMQO24(288)
!           L=1
!           DO WHILE (dailyq.gt.0.d0.and.L.le.213)
!                dqdt=arec*dailyq**brec
!                IF (dqdt.ge.dailyq) then
!                     dqdt=dailyq
!                END IF
!                DBASEF(JDAY+L,INODE)=dailyq-dqdt +
DBASEF(JDAY+L,INODE)
!                dailyq=dailyq-dqdt
!                L=L+1
!
PRINT*,'baseflow',jday,inode,dbasef(jday+L,inode)
!            END DO
!
!          CALL bfcalc(SISTORE,DBASEF,DRECH,DSM2,DTHETA2,INODE,
!     C          dstorvol,dmaxstor,drloss,dSIWater1,dminstor)
!          END IF
!      END IF
!

C ----- SEDIMENT MOVEMENT CALCULATIONS -----
      IF (Q.GT.0.D0) THEN
          IF (ieroty.eq.1)THEN !!     1) Williams (1975)
              DSED(JDAY,INODE)=SCONC1
          ELSEIF (ieroty.eq.2)THEN !! 2) GLEAMS / daily CREAMS
              DSED(JDAY,INODE)=SCONC2
          ELSEIF (ieroty.eq.3)THEN !! 3) Foster et al. (1977)
              DSED(JDAY,INODE)=SCONC
```

```
          ELSEIF (ieroty.eq.4)THEN !! 4) Cooley (1980) - Design
Storm
              DSED(JDAY,INODE)=SCONC3
          ENDIF
      ELSE
          DSED(JDAY,INODE)=0.D0
      END IF
      !print*,'dsed',Dsed(jday,inode)
c --- STORE STAIL FOR NEXT DAY ---
C ----------------------------------------------------------------
-----------------
C    OUTPUT - FORMAT
C ----------------------------------------------------------------
-----------------
1000  FORMAT(6X,'NO NEW FLOW GENERATED ON THIS DAY')

      RETURN
      END SUBROUTINE uhcn
```

```
C --------------------------------------------------------------
------------------------------
C   Program:
C --------------------------------------------------------------
------------------------------
      SUBROUTINE unit_hyd(Q,Area,qp,tp,D,tc,mref)
C --------------------------------------------------------------
------------------------------
C Unit NRCS hydrograph using Haan's equation (k=3.77)
C --------------------------------------------------------------
------------------------------
C      version 3.0.1, Last ModIFied: See ModIFications below
C      WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C      Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons,
BAE (jep)
C                  University of Florida        BAE, NC State
University
C                  Gainesville, FL 32611        Raleigh, NC
27695-7625(USA)
C                  e-mail: carpena@ufl.edu
C --------------------------------------------------------------
------------------------------
C   DEFINE VARIABLES
C Q: Runoff total, mm !!!! says cm in documentation
C Area: Watershed area, ha
C qp: Peak flow, m3/s
C tp: Time to peak flow, minutes
C D: Storm duration, h
C tc: time of concentration, hours
C qp5: 5-minute unit hydrograph peak flow, m3/s
C tp5: 5-minute unit  hydrograph time to peak, hr
C mref: number of unit hydrograph steps
C u(5000,2):matrix holding time (t5) in column 1 and flow (qi5,
m3/s) in column 2
C --------------------------------------------------------------
------------------------------
C   DECLARE VARIABLES
C --------------------------------------------------------------
------------------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))
      COMMON/hydgph/u(5000,2),qh(5000,3)

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
C --------------------------------------------------------------
------------------------------
      ck=3.77d0
```

```
      t5=0.d0 !MAC 10/04/12
C --------------------------------------------------------------
------------------------------
C  INITIALIZE u VECTOR
C --------------------------------------------------------------
------------------------------
      DO 5 i=1,5000
         DO 5 j=1,2
            u(i,j)=0.d0
5     CONTINUE

C ---rmc 09/15/11- New total hydrograph calculation from unit
hydrograph
C ---unit hidrograph values.
Def:duration(h),qp5(m3/s),tp5(h):peak t,q
C ---a)Estimate time step for dimesionless unit hydrograph as
Def<>1/3.tp
C ---since tp=0.6tc+Def/2 --> IF Def<>1/3tp --> Def<>0.24tc
      Def=0.24d0*tc !MAC 04/10/12 Time step defined as 5 min
      !WRITE(2,205)Def*60.d0 !MAC 04/10/12
      tp5=0.6d0*tc+0.5d0*Def !time to peak in hours
      qp5=0.127481d0*Area/(tp5*60.d0) !m3/s
      !print*,'area,tp5,tc,def',Area,tp5,tc,def
      qdepth=qp5*360.d0/Area !peak flow converted to mm/h
      WRITE(NUT,204)5.d0
      WRITE(NUT,1000)qp5,qdepth
      WRITE(NUT,1100)tp5,tp5*60.d0
c      WRITE(NUT,205)
C --------------------------------------------------------------
-------------------------------
C -- SCS TRIANGULAR HYDROGRAPH
C --------------------------------------------------------------
-------------------------------
c     ttotal5=2.67d0*tp5
C --------------------------------------------------------------
-------------------------------
C ---SCS aDIMENSIONLESS UNITH HYDROGRAPH
      ttotal5=5.d0*tp5
C --------------------------------------------------------------
-------------------------------
      dt5=5.d0/60.d0
      i=0
      cqdepth5=0.d0

      DO WHILE (t5.le.ttotal5)
         t5=i*dt5
         IF(Q.le.0) THEN
```

```
            qi5=0.d0
         ELSE
           qi5=qp5*((t5/tp5*dexp(1-t5/tp5))**ck)
         END IF
         u(i+1,1)=t5
         u(i+1,2)=qi5
         qdepth5=qi5*360.d0/Area !Explanation, to obtain q in
(mm/h)
         cqdepth5=qdepth5+cqdepth5
c         WRITE(NUT,110)(u(i+1,j),j=1,2),qdepth5
         i=i+1
      END DO
C ---rmc - mref, number of unit hydrograph steps needed in
convolution
      mref=i
C ---rmc - check unit hydrograph volumen <> 1
      unitq=cqdepth5*dt5
      IF(dabs(1.d0-unitq).le.0.05d0) THEN
         WRITE(NUT,120)'-->PASSED unit hydrograph check-
V(mm)=',unitq
      ELSE
         WRITE(NUT,120)'-->FAILED unit hydrograph check-
V(mm)=',unitq
      END IF
C ---------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C ---------------------------------------------------------------
----------
1000  FORMAT(4X,'Peak flow    =',f9.3,' m3/s = ',f9.4,' mm/h')
1100  FORMAT(4X,'Time to peak =',f8.2,' h    = ',f8.2,' min')
204   FORMAT(4X,'a) SCS ',f4.2,' - min UNIT HYDROGRAPH:')
205   FORMAT(/,4X,' Time (h)    q(m3/s)   q(mm/h)',/,4x,
     C  30('-'))
100   FORMAT(f9.2,2f10.4)
110   FORMAT(3f10.4)
120   FORMAT(4X,A38,f6.2)

      RETURN
      END SUBROUTINE unit_hyd
```

```
! Program 12 - Based on Hromadka book pg. 157
C  ---------------------------------------------------------------
-------------------------
! SUBROUTINES of nine processes considerer in (Hromadka et al.,
1983)
! Each of one is called from java, and at this moment the SS
array is
! returned (UF, 10/5/2018- Marco Pazmiño-Hernandez modified)
C ----------------------------------------------------------------
-------------------------
      SUBROUTINE unith(m,n,m1,n1,mn1,mn2) ! ARGU = nut (8.29.18)
C ----------------------------------------------------------------
-------------------------
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCC
C NA:      Stream "A" number. This stream is the one to be
modeled                        C
C KTYPE:   Select 24-hr storm unit-interval model number
                              C
C XL:      Piper length - the length of the  longest watercourse
(FEET)                        C
C XLCA:    The length along the longest watershed watercourse
measured from              C
C          the point of concentration upstream to a point
opposite the                    C
C          centroid of the watershed area
C
C HH:      The difference in elevation between the most report
point in the               C
C          watershed and the point of concentration (FEET)
C
C XN:      Basin Factor (Manning's Friction Factor) [ 0.008 -
0.999]                       C
C AREA:
C
C VSL:     Lost rate (inch/hour)
C
C KODE1:   Unit-Hydograph "S" graph options: 1. Valley zone, 2.
Foothill Zone,           C
C          3. Mountain Zone, 4. Desert Zone, 5. Combination of
option 1 to 4              C
C BASCON:  BASEFLOW (CFS/square-mile)
C
C SLP:     Low lost rate percentage (decimal notation)
C
C R5:      5 Min [inches]     - Watershed    area-averaged point
rainfalls                       C
```

```
C R30:   30 Min
C
C R1:      1 Hour
C
C R3:      3 Hour
C
C R6:      6 Hour
C
C R24:    24 Hour
C
C SS*
C
C KSTORM*
C
C KSOIL: Efective rainfall information display options
C
C PV:      Percentage (decimal notation) of watershed specified
with Valley "S" curve      C
C PF:      Percentage (decimal notation) of watershed specified
with Foothill "S" curve    C
C PM:      Percentage (decimal notation) of watershed specified
with Mountain "S" curve    C
C PD:      Desert "S" curve percentage
C
C NUT*
C
C FX5:     5 Min - Depth-Area Adjustrent Factor
C
C FX30:   30 min
C
C FX1:     1 Hour
C
C FX3:     3 Hour
C
C FX6:     6 Hour
C
C FX24: 24 Hour
C
C IDAOPT: User-specified depht-area factor
C
C TIME1: Time for Beginning of results (hrs)
C
C TIME2: Time for End of results (hrs)
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
C -------------------------------------------------------------
-------------------------
C   DECLARE VARIABLES
C -------------------------------------------------------------
-------------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      DIMENSION H(5555)
      DIMENSION UH(150),PERCNT(150),R(288)
      DIMENSION DYR(48)
      DATA
DYR/0.,20.,40.,60.,80.,100.,125.,150.,100.,89.4,82.3,78.,
     C
74.8,72.4,70.2,68.5,100.,94.8,91.2,88.4,86.3,84.5,82.7,81.3,
     C
100.,96.2,93.8,91.8,90.3,89.,87.8,86.7,100.,97.4,95.8,94.8,
     C 93.9,93.2,92.8,92.3,100.,82.,72.5,66.6,63.,60.8,59.,57.5/
      DATA UH/150*0.d0/
      DATA PERCNT/150*0.d0/
      DATA R/288*0.d0/
!      DATA H/5555*0.d0/
!      Hydro RETURNs the hydrograph, Time(Hours) and
Discharge(CFS)
C -------------------------------------------------------------
------------------------

READ(nut,*)NA,KTYPE,XL,XLCA,HH,XN,AREA,VSL,KODE1,BASCON,SLP,
     C  KSOIL,PV,PF,PM,PD,IDAOPT,TIME1,TIME2
      READ(NIPR,*)NZ1R,NZ2R,KODER
      IF(NZ1R.EQ.NZ1.AND.NZ2R.EQ.NZ2.AND.KODER.EQ.KODE) THEN

READ(NIPR,*)R5,R30,R1,R3,R6,R24,FX5,FX30,FX1,FX3,FX6,FX24
      END IF
C -------------------------------------------------------------
------------------------
      !SS=SS1 ! M.P.
      AX=AREA
      XLX=XL
      XLCAX=XLCA
C OPO This line was added to clear the hydrograph H
C before the next hydrograph in another node.
      !DO 15 I=1,440
         H(I)=0.d0
!15    CONTINUE
c rmc FX=1.
```

```
      IF(IDAOPT.EQ.2)GO TO 183
         AREA=AREA/640.d0
         FX1=.651d0
         FX3=.78d0
         FX6=.831d0
         FX24=.91d0
         IF(AREA.GE.350.d0)GO TO 180
         DO 100 I=1,13
            IF(AREA.LE.DYR(I))GO TO 110
100         CONTINUE
110         I=I-1
C --------------------------------------------------------------
-------------------------
            DX=DYR(I+1)-DYR(I)
            FACT=AREA-DYR(I)
            FX1=(DYR(I+8)-FACT*(DYR(I+8)-DYR(I+9))/DX)/100.d0
            FX3=(DYR(I+16)-FACT*(DYR(I+16)-DYR
(I+17))/DX)/100.d0
            FX6=(DYR(I+24)-FACT*(DYR(I+24)-DYR(I+25))/DX)/100.d0
            FX24=(DYR(I+32)-FACT*(DYR(I+32)-
DYR(I+33))/DX)/100.d0
            FX30=(DYR(I+40)-FACT*(DYR(I+40)-
DYR(I+41))/DX)/100.d0
180      CONTINUE
         AREA=AREA*640.d0
183   CONTINUE
C --------------------------------------------------------------
-------------------------
      WRITE(NUT,5000)
      WRITE(NUT,181)
      WRITE(NUT,5000)
C --------------------------------------------------------------
-------------------------
C  DETERMINE HYDROGRAPH FACTORS
C --------------------------------------------------------------
-------------------------
      XL=XL/5280.d0
      XLCA=XLCA/5280.d0
      S=HH/XL
      XLAGX=1.2d0*(XL*XLCA/S**0.5d0)**0.38d0
      XLAG=20.d0*XN*XLAGX
C --------------------------------------------------------------
-------------------------
C  DESIGNATE UNIT INTERVALS [UNIT in minutes]
C --------------------------------------------------------------
-------------------------
      IF(KTYPE.EQ.1)UNIT=5.d0
```

```
      IF(KTYPE.EQ.2)UNIT=10.d0
      IF(KTYPE.EQ.3)UNIT=15.d0
      IF(KTYPE.EQ.4)UNIT=20.d0
      IF(KTYPE.EQ.5)UNIT=30.d0
      IF(KTYPE.EQ.6)UNIT=60.d0
C ---------------------------------------------------------------
-------------------------
      TIMLAG=100.d0*UNIT/60.d0/XLAG
      SQMI=AREA*43560.d0/5280.d0/5280.d0
      BASFLO=BASCON*SQMI
      XK=645.d0*SQMI*60.d0/UNIT
      XK=XK/100.d0
C ---------------------------------------------------------------
-------------------------
C  RETURN XL AND XLCA TO UNITS OF FEET
C ---------------------------------------------------------------
-------------------------
      XL=XL*5280.d0
      XLCA=XLCA*5280.d0
C ---------------------------------------------------------------
-------------------------
      WRITE(NUT,5010)
      WRITE(NUT,5001)XLX,XLCAX,HH,XN,AX,XLAG,UNIT,TIMLAG,
     C BASFLO,VSL,SLP
C ---------------------------------------------------------------
-------------------------
      IF(KODE1.EQ.1)WRITE(NUT,8200)
      IF(KODE1.EQ.2)WRITE(NUT,8202)
      IF(KODE1.EQ.3)WRITE(NUT,8204)
      IF(KODE1.EQ.4)WRITE(NUT,8205)
      IF(KODE1.EQ.5)WRITE(NUT,82041)PV,PF,PM,PD
      IF(KODE1.EQ.6)WRITE(NUT,82042)
C ---------------------------------------------------------------
-------------------------
      WRITE(NUT,8209)R5,R30
      WRITE(NUT,8206)R1,R3,R6,R24,KTYPE
C ---------------------------------------------------------------
-------------------------
      IF(IDAOPT.EQ.1)WRITE(NUT,55555)
      IF(IDAOPT.EQ.2)WRITE(NUT,55556)
      WRITE(NUT,55554)FX5,FX30,FX1,FX3,FX6,FX24
      WRITE(NUT,5000)
C ---------------------------------------------------------------
-------------------------
C  SOIL INFILTRATION EFFECTIVE LOSS RATE, XR
C ---------------------------------------------------------------
-------------------------
```

```
      XR=VSL*UNIT/60.d0
C --------------------------------------------------------------
-------------------------
C  DETERMINATION OF UNIT HYDROGRAPH ORDINATES
C --------------------------------------------------------------
-------------------------
      IF(KODE1.LT.5 .OR. KODE1.EQ.6)GO TO 94
C --------------------------------------------------------------
-------------------------
C  LINEAR WEIGHTING OF S-CURVES
C --------------------------------------------------------------
-------------------------
      NLAG=0
      DO 98 KLAG=1,4
         PLAG=PV
         IF(KLAG.EQ.2)PLAG=PF
         IF(KLAG.EQ.3)PLAG=PM
         IF(KLAG.EQ.4)PLAG=PD
         IF(KLAG.EQ.0)GO TO 98
         CALL SUBSB(TIMLAG,PERCNT,KLAG,NUMBER)
         PRINT*,'PERCNT=',PERCNT
         IF(NUMBER.GT.NLAG)NLAG=NUMBER
         DO 99 I=1,150
            IF (PERCNT(I).EQ.0.d0)PERCNT(I)=100.d0
            H(I)=H(I)+PERCNT(I)*PLAG
             PRINT*,'H=',H
!!99           PERCNT(I)=0.d0
              PERCNT(I)=0.d0
99       CONTINUE
98    CONTINUE
C --------------------------------------------------------------
-------------------------
C  UNIT HYDROGRAPH DETERMINATION
C --------------------------------------------------------------
-------------------------
      NUMBER=NLAG
      DO 97 I=1,150
         IF(I.LE.NLAG)PERCNT(I)=H(I)
!!97           H(I)=0.d0
            H(I)=0.d0
97    CONTINUE
      GO TO 96
94    CALL SUBSB(TIMLAG,PERCNT,KODE1,NUMBER)
!94    CALL SUBSB(TIMLAG,PERCNT,KODE1,NUMBER,NUT)
96    SUM=0.d0
C --------------------------------------------------------------
-------------------------
```

```
      IF(NUMBER.GE.150)WRITE(NUT,8207)

      IF(NUMBER.GT.150)NUMBER=150
C
      DO 8208 I=1,NUMBER
         UH(I)=(PERCNT(I)-SUM)*XK
         SUM=PERCNT(I)
         IF(UH(I).LT.0.d0)UH(I)=0.d0
8208  CONTINUE
C ---------------------------------------------------------------
-------------------------
      WRITE(NUT,5010)
      WRITE(NUT,5002)
      WRITE(NUT,5010)
      WRITE(NUT,5004)
      WRITE(NUT,5010)
      WRITE(NUT,5005)(I,PERCNT(I),UH(I),I=1,NUMBER)
      WRITE(NUT,5010)
C ---------------------------------------------------------------
-------------------------
C 24-HOUR STORM RAINFALL PATTERN
C ---------------------------------------------------------------
-------------------------
      R5A=R5*FX5
      R30A=R30*FX30
      R1A=R1*FX1
      R3A=R3*FX3
      R6A=R6*FX6
      R24A=R24*FX24
C ---------------------------------------------------------------
-------------------------
      A=(DLOG(R30A)-DLOG(R5A))/(DLOG(.5d0)-DLOG(.0833d0))
      B=DLOG(R5A)-A*DLOG(.0833d0)
      R(193)=RR(A,B,.0833d0)
      R(194)=RR(A,B,.1667d0)-RR(A,B,.0833d0)
      R(195)=RR(A,B,.25d0)-RR(A,B,.1667d0)
      R(192)=RR(A,B,.3333d0)-RR(A,B,.25d0)
      R(196)=RR(A,B,.41667d0)-RR(A,B,.3333d0)
      R(191)=RR(A,B,.5d0)-RR(A,B,.41667d0)
C ---------------------------------------------------------------
-------------------------
      A=(DLOG(R1A)-DLOG(R30A))/(DLOG(1.d0)-DLOG(.5d0))
      B=DLOG(R30A)-A*DLOG(.5d0)
      R(197)=RR(A,B,.5833d0)-RR(A,B,.5d0)
      R(190)=RR(A,B,.55557d0)-RR(A,B,.5833d0)
      R(198)=RR(A,B,.75d0)-RR(A,B,.55557d0)
      R(189)=RR(A,B,.8333d0)-RR(A,B,.75d0)
```

```
      R(188)=RR(A,B,.9167d0)-RR(A,B,.8333d0)
      R(187)=RR(A,B,1.d0)-RR(A,B,.9167d0)
C ----------------------------------------------------------------
-------------------------
C  REMAINING PART OF PEAK 3-HOUR STORM
C ----------------------------------------------------------------
-------------------------
      A=(DLOG(R3A)-DLOG(R1A))/(DLOG(3.d0)-DLOG(1.d0))
      B=DLOG(R1A)-A*DLOG(1.d0)
      RRSAVE=R1A
C ----------------------------------------------------------------
-------------------------
      DO 1001 J=1,12
         XJ=J
         DT=XJ*.1667d0
         T=1.d0+DT !first time that T is mentioned
         RRNEW=RR(A,B,T)
         DR=(RRNEW-RRSAVE)/2.d0
         R(J+198)=DR
         IR=187-J
         R(IR)=DR
!!1001     RRSAVE=RRNEW
         RRSAVE=RRNEW
1001  CONTINUE
C ----------------------------------------------------------------
-------------------------
C  REMAINING PART OF PEAK 6-HOUR STORM
C ----------------------------------------------------------------
-------------------------
      A=(DLOG(R6A)-DLOG(R3A))/(DLOG(6.d0)-DLOG(3.d0))
      B=(DLOG(R3A)-(A*DLOG(3.d0)))
      RRSAVE=R3A
C ----------------------------------------------------------------
-------------------------
      DO 1010 J=1,18
         XJ=J
         DT=XJ*.1667d0
         T=3.d0+DT
         RRNEW=RR(A,B,T)
         DR=(RRNEW-RRSAVE)/2.d0
         R(J+210)=DR
         IR=175-J
         R(IR)=DR
!!1010  RRSAVE=RRNEW
         RRSAVE=RRNEW
1010  CONTINUE
```

```
C -----------------------------------------------------------------
-------------------------
C  REMAINING PART OF PEAK 24-HOUR STORM
C -----------------------------------------------------------------
-------------------------
      A=(DLOG(R24A)-DLOG(R6A))/(DLOG(24.d0)-DLOG(6.d0))
      B=DLOG(R6A)-A*DLOG(6.d0)
      RRSAVE=R6A
C -----------------------------------------------------------------
-------------------------
      DO 1020 J=1,60
         XJ=J
         DT=XJ*.1667d0
         T=6.d0+DT
         RRNEW=RR(A,B,T)
         DR=(RRNEW-RRSAVE)/2.d0
         R(J+228)=DR
         IR=157-J
         R(IR)=DR
!!1020     RRSAVE=RRNEW
         RRSAVE=RRNEW
1020  CONTINUE
C -----------------------------------------------------------------
-------------------------
      DO 1030 J=1,96
         XJ=J
         DT=XJ*.08333d0
         T=16.d0+DT
         RRNEW=RR(A,B,T)
         DR=RRNEW-RRSAVE
         IR=97-J
         R(IR)=DR
!!1030     RRSAVE=RRNEW
         RRSAVE=RRNEW
1030  CONTINUE
C -----------------------------------------------------------------
-------------------------
C  ADJUST R-ARRAY FOR LARGER UNIT INTERVALS
C -----------------------------------------------------------------
-------------------------
C  NI = NUMBER OF STORM HYDROGRAPH INTERVALS
C -----------------------------------------------------------------
-------------------------
      NI=288
      IF(KTYPE.EQ.1)GO TO 1050
         K=KTYPE
      IF(KTYPE.EQ.5)K=6
```

```
      IF(KTYPE.EQ.6)K=12
         NI=288/K
         DO 1040 I=1,NI
            TEMP=0.d0
            II=(I-1)*K
            DO 1035 J=1,K
               IR=II+J
!!1035        TEMP=TEMP+R(IR)
!!1040        R(I)=TEMP
               TEMP=TEMP+R(IR)
1035        CONTINUE
            R(I)=TEMP
1040     CONTINUE
1050  CONTINUE
C        ADJUST FOR CONSTANT SOIL LOSS
         XTOTAL=0.d0
c rmc   TEMPS=UNIT/60.
         XRA=XR
C --------------------------------------------------------------
-------------------------
      IF(KSOIL.EQ.1)WRITE(NUT,5000)
      IF(KSOIL.EQ.1)WRITE(NUT,5020)
      IF(KSOIL.EQ.1)WRITE(NUT,5003)

      DO 300 I=1,NI
         XLOSS=R(I)*SLP
         IF(XLOSS.GT.XRA) XLOSS=XRA
            TEMP=R(I)
            R(I)=R(I)-XLOSS
            XTOTAL=XTOTAL+XLOSS
C --------------------------------------------------------------
-------------------------
            IF(KSOIL.EQ.1)WRITE(NUT,5021)I,TEMP,XLOSS,R(I)
300   CONTINUE
C --------------------------------------------------------------
-------------------------
!      DO 14 I=1,288
!           R(I)=PP(I,2)/25.4 !Change of unit, from mm to inch
!14    CONTINUE
!      NI=PP(500,1)
C --------------------------------------------------------------
-------------------------
C DETERMINE STORM RUNOFF HYDROGRAPH
C --------------------------------------------------------------
-------------------------
      INTERV=NUMBER+NI-1
```

```
C ---------------------------------------------------------------
-------------------------
      IF(INTERV.GT.440)WRITE(NUT,432)
         IF(INTERV.GT.439)THEN
          INTERV=439
          WRITE(*,*) 'RUNOFF HYDROGRAPH TERMINATED AFTER 440
UNIT'
          WRITE(*,*) 'INTERVALS. SUGGEST USING A LARGER UNIT
INTERVAL'
      END IF
C ---------------------------------------------------------------
-------------------------
      DO 600 I=1,INTERV
         M=I
         N=1+M
         DO 500 J=1,M
            K=N-J
            IF(J.GT.NI)GO TO 500
            IF(K.GT.NUMBER) GO TO 500
               H(M)=H(M)+R(J)*UH(K)
500      CONTINUE
         H(M)=H(M)+BASFLO
600   CONTINUE
C ---------------------------------------------------------------
-------------------------
C COMPUTE SUMMED HYDROGRAPH
C ---------------------------------------------------------------
-------------------------
      SUM=0.d0
      XMAX=0.d0
      DO 700 I=1,INTERV
         IF(H(I).LT.0.d0)H(I)=0.d0
         IF(H(I).GT.XMAX)XMAX=H(I)
            SUM=SUM + H(I)
 700  CONTINUE
      SUM=SUM*UNIT*60.d0/43560.d0
      XTOTAL=XTOTAL/12.d0*AREA
C ---------------------------------------------------------------
-------------------------
     WRITE(NUT,5003)
     WRITE(NUT,6008)XTOTAL,SUM
     IF(XMAX.LT.100.d0)GO TO 8000
        I=INT(XMAX/100.d0)
        II=I+1
        XMAX=II
        XMAX=XMAX*100.d0
        GO TO 8100
```

```
8000    I=INT(XMAX/10.d0)
        II=10*(I+1)
        XMAX=II
8100  CONTINUE
C --------------------------------------------------------------
-------------------------
       WRITE(NUT,5003)
C --------------------------------------------------------------
-------------------------
C Pass hydrograph to STREAM flow MATRIX SS
C --------------------------------------------------------------
-------------------------
      CALL ADDHY(UNIT,INTERV,NA,H)
C --------------------------------------------------------------
-------------------------
C Print results and estimate mass balances
C --------------------------------------------------------------
-------------------------
      CALL OASB(KTYPE,H,INTERV,XMAX,UNIT,SUM,TIME1,TIME2)
!        CALL ADDHY(UNIT,INTERV,NA,H)
C --------------------------------------------------------------
----------
C  OUTPUT - FORMAT
C --------------------------------------------------------------
----------
181   FORMAT(/,28X,'UNIT-HYDROGRAPH ANALYSIS',/)
55556  FORMAT(/,11X,'USER SPECIFIED PRECIPITATION DEPTH-AREA',
     C   ' REDUCTION FACTORS:')
55555  FORMAT(/,11X,'PRECIPITATION DEPTH-AREA REDUCTION
FACTORS:')
55554  FORMAT(24X,'5 - MINUTE FACTOR = ',F12.3,/,
     C 23X,'30 - MINUTE FACTOR = ',F12.3,/,
     C 24X,'1 -  HOUR  FACTOR = ',F12.3,/,24X,'3 -  HOUR  FACTOR
= ' ,
     C F12.3,/,24X,'6 -  HOUR  FACTOR = ',F12.3,/,23X,'24 -
HOUR  FACTO
     CR = ',F12.3,/)
5001  FORMAT(11X,'WATERCOURSE LENGTH = ',F37.3,' FEET',/,
     C 11X,'LENGTH FROM CONCENTRATION POINT TO CENTROID = ',
     C F12.3,' FEET',/,11X,'ELEVATION VARIATION ALONG
     C WATERCOURSE = ',F18.3,' FEET',/,11X,'MANNINGS
     C FRICTION FACTOR ALONG WATERCOURSE = ',F13.3,/,
     C 11X,'WATERSHED AREA = ',F41.3,' ACRES',
     C /,11X,'WATERCOURSE "LAG" TIME = ',F33.3,' HOURS',/,
     C 11X,'UNIT HYDROGRAPH TIME UNIT = ',F30.3,' MINUTES',/,
     C 11X,'UNIT INTERVAL PERCENTAGE OF LAG-TIME = ',F19.3,/,
     C 11X,'HYDROGRAPH BASEFLOW = ',F36.3,' CFS',/,11X,
```

```
     C 'UNIFORM MEAN SOIL-LOSS(INCH/HOUR) = ',F22.3,/,
     C 11X,'LOW SOIL-LOSS RATE PERCENT(DECIMAL) = ',F20.3)
5000  FORMAT(1X,76('*'))
5010  FORMAT(1X,76('='))
5020  FORMAT(11X,'UNIT',14X,'UNIT',12X,'UNIT',14X,'EFFECTIVE',/,
     C 10X,'PERIOD',11X,'RAINFALL',7X,'SOIL-
LOSS',12X,'RAINFALL',/,
     C
9X,'(NUMBER)',10X,'(INCHES)',8X,'(INCHES)',12X,'(INCHES)')
5021  FORMAT(11X,I3,12X,F7.4,10X,F7.4,13X,F7.4)
82041 FORMAT(11X,'VALLEY "S"-CURVE PERCENTAGE(DECIMAL NOTATION)
= ',
     C F6.3,/,11X,'FOOTHILL "S"-CURVE PERCENTAGE(DECIMAL
NOTATION) = ',
     C F6.3,/,11X,'MOUNTAIN "S"-CURVE PERCENTAGE(DECIMAL
NOTATION) = ',
     C F6.3,/,11X,'DESERT "S"-CURVE PERCENTAGE(DECIMAL NOTATION)
= ',
     C F6.3,/)
8205  FORMAT(11X,'DESERT S-GRAPH SELECTED',/)
8200  FORMAT(11X,'VALLEY S-GRAPH SELECTED',/)
8202  FORMAT(11X,'FOOTHILL S-GRAPH SELECTED',/)
8204  FORMAT(11X,'MOUNTAIN S-GRAPH SELECTED',/)
82042 FORMAT(10X,'U.S. SOIL CONSERVATION SERVICE S-GRAPH
SELECTED',/)

8209  FORMAT(11X,'SPECIFIED PEAK  5-MINUTES RAINFALL(INCH) =
',F15.2,/,
     C 11X,'SPECIFIED PEAK 30-MINUTES RAINFALL(INCH) = ',F15.2)

8206  FORMAT(11X,'SPECIFIED PEAK  1-HOUR RAINFALL(INCH)    =
',F15.2,/,
     C 11X,'SPECIFIED PEAK  3-HOUR RAINFALL(INCH)    =
',F15.2,/,
     C 11X,'SPECIFIED PEAK  6-HOUR RAINFALL(INCH)    =
',F15.2,/,
     C 11X,'SPECIFIED PEAK 24-HOUR RAINFALL(INCH)    =
',F15.2,//,
     C 18X,'    HYDROGRAPH MODEL # ',I1, 1X '    SPECIFIED*')
8207  FORMAT(4X,'UNIT HYDROGRAPH TERMINATED AFTER 150 UNIT
INTERVALS',/)
6008  FORMAT(6X,'TOTAL SOIL-LOSS VOLUME(ACRE-FEET)    =
',F28.4,/,
     C 6X,'TOTAL STORM RUNOFF VOLUME(ACRE-FEET) = ',F28.4)
5002  FORMAT(/,26X,'UNIT HYDROGRAPH DETERMINATION',/)
5003  FORMAT(1X,76('-'))
```

```
5004  FORMAT(6X,'INTERVAL',12X,'"S" GRAPH',12X,'UNIT
HYDROGRAPH',/,
     C 7X,'NUMBER',12X,'MEAN VALUES',12X,'ORDINATES(CFS)')
5005  FORMAT(8X I3,15X,F7.3,13X,F10.3)
432   FORMAT(4X,'RUNOFF HYDROGRAPH TERMINATED AFTER 440 UNIT',/,
     C 4X,'INTERVALS. SUGGEST USING A LARGER UNIT INTERVAL.')
C -----------------------------------------------------------------
------------------------
C HYDROGRAPH TO EXPORT
C -----------------------------------------------------------------
------------------------
!      Hydro=H
!      write(*,*) INTERV
!      write(*,*) "Hydrograph UNITH"
!      DO 715 I=1,440
!         Hydro(I,2)=H(I)/(0.3048**3) !To obtain hydro in m^3/s
!          Hydro(I,2)=H(I) !hydro in CFS
!         TIMEOUT=TIMEOUT+.083333d0
!         WRITE(*,*) TIMEOUT
!         WRITE(*,*) Hydro(I)
!715   CONTINUE
!      DO 716 I=1,mn1
!         IF(I==1) THEN
!            Hydro(I,1)=0.083333d0
!         ELSE
!            J=I-1
!            Hydro(I,1)=Hydro(J,1)+0.083333d0
!         END IF
!         WRITE(*,*) Hydro(I,1), Hydro(I,2)
!716   CONTINUE
!      WRITE(*,*) Hydro(1,1), Hydro(1,2)
!      SS1=SS ! Just to update its value
!10000  CONTINUE
       RETURN
       END SUBROUTINE unith


C -----------------------------------------------------------------
-----------------------
C RR - FUNCTION outside of the UNITH subroutine
C -----------------------------------------------------------------
-----------------------
      FUNCTION RR(A,B,T)
C -----------------------------------------------------------------
-----------------------
      IMPLICIT DOUBLE PRECISION (a-h, o-z)
         RR = DEXP((A*DLOG(T))+B)
```

```
RETURN
END FUNCTION RR
```

```
C ---------------------------------------------------------------
----------------------------------------------------------------
------------------------------------------------------
C   Program:
C---------------------------------------------------------------
      SUBROUTINE
vfsout(dp,ieroty,sconc,sconc1,sconc2,Area,pL,qp,tp,
     C                  tc,D,ti,nhyet,nhyd)
C---------------------------------------------------------------
C   Version 3.0.1, Last ModIFied: See ModIFications below
C   WRITTEN FOR: ASAE'99 Toronto paper, March 8, 2002
C   Written by: R. Munoz-Carpena (rmc)   &   J. E. Parsons, BAE
(jep)
C              University of Florida        BAE, NC State
University
C              Gainesville, FL 32611        Raleigh, NC 27695-
7625(USA)
C             e-mail: carpena@ufl.edu
C---------------------------------------------------------------
C---------------------------------------------------------------
C OUTPUT FOR VFSMOD INPUT FILES
C---------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (a-h,o-z)
C      PARAMETER (5555=INT(600))

COMMON/NINOUT/NUT,NDAT,NIPR,NSSS,NDSS,NZ1,NZ2,JCOUNT,JDAY,KODE
      COMMON/hydgph/u(5000,2),qh(5000,3)

COMMON/rain/rfix,rti(5000),rfi(5000),rcum(5000,2),ref(5000,2),nc
um
C---------------------------------------------------------------
C OUTPUT OF VFSMOD INPUT FILE: *.isd
C---------------------------------------------------------------
       npart=7
       coarse=1.0d0
      IF (ieroty.eq.1) THEN
         ci= sconc1/1000.d0
      ELSE IF (ieroty.eq.2) THEN
         ci= sconc2/1000.d0
      ELSE IF (ieroty.eq.3) THEN
         ci= sconc/1000.d0
      ELSE
         ci= sconc1/1000.d0
      END IF
c--rmc  05/08/03 when runoff is small, sediment concentration by
sediment
```

```
c----- yields methods that DO not consider runoff in calculation
(Foster's
c----- ,CREAMS) can be very large. Override user selection of
the method
c----- and slect Williams' that considers runoff and typiCALLy
avoids this
c----- problem. Issue warning.
     !IF(ci.ge.0.25d0) THEN
 !     ci=sconc1/1000.d0
 !     WRITE(*,160)
 !     WRITE(*,*)'WARNING: small runoff in this case produces
large',
 !    C    ' sediment concentration with the sediment yield
method #',
 !    C    ieroty,'selected. Using Williams method instead--see
manual'
 !     WRITE(*,160)
 !     WRITE(10,*)
 !     WRITE(10,160)
 !     WRITE(10,*)'WARNING: small runoff in this case produces
large',
 !    C    ' sediment concentration with the sediment yield
method #',
 !    C    ieroty,'selected. Using Williams method instead--see
manual'
 !     WRITE(10,160)
 !     END IF
     !por=0.434d0
     !WRITE (15,101) Npart,coarse,ci,por
     !dpp=dp/10000.d0
     !sg=2.65d0
     !WRITE (15,102)dpp,sg
C -------------------------------------------------------------
      IF(ci.ge.0.25d0) THEN
         ci=sconc1/1000.d0
         WRITE(NUT,160)
         WRITE(NUT,*)'WARNING: small runoff in this case
produces large'
     C   ,' sediment concentration with the sediment yield
method #',
     C    ieroty,'selected. Using Williams method instead--see
manual'
         WRITE(NUT,160)
         WRITE(NUT,*)
         WRITE(NUT,160)
         WRITE(NUT,*)'WARNING: small runoff in this case
produces large'
```

```
     C   ,' sediment concentration with the sediment yield
method #',
     C    ieroty,'selected. Using Williams method instead--see
manual'
         WRITE(NUT,160)
      END IF
     por=0.434d0
     WRITE(NUT,101) Npart,coarse,ci,por
     dpp=dp/10000.d0
     sg=2.65d0
     WRITE(NUT,102)dpp,sg
C-------------------------------------------------------------------
C OUTPUT of VFSMOD runoff hydrograph: *.iro
C-------------------------------------------------------------------
     swidth=Area*10000.d0/pL
     slength=pL
!    WRITE (12,103) swidth,slength
     WRITE(NUT,103) swidth,slength
     nbcroff=nhyd
     bcropeak=qp
      nstep1=100
      IF(nhyd.le.nstep1)THEN
!      WRITE (12,104)nbcroff+1,bcropeak
!      WRITE(2,*)' '
!      WRITE(2,250)ti
         WRITE(NUT,104)nbcroff+1,bcropeak
         WRITE(NUT,*)' '
         WRITE(NUT,250)ti
      DO 20 ii=1, nbcroff-1
         tt=qh(ii,1)*3600.d0
      IF (ii.eq.1) THEN
!         WRITE(12,105)tt,qh(ii,2)
         WRITE(NUT,105)tt,qh(ii,2)
      ELSE
!         WRITE(12,106)tt,qh(ii,2)
         WRITE(NUT,106)tt,qh(ii,2)
      END IF
20    CONTINUE
      ELSE
      nWRITE1=nhyd/nstep1+1
!      WRITE(12,104)nhyd/nWRITE1+2,bcropeak
       WRITE(NUT,104)nhyd/nWRITE1+2,bcropeak
      DO 29 ii=1,nhyd-1
         tt=qh(ii,1)*3600.d0
      IF (ii.eq.1) THEN
 !       WRITE(12,105)tt,qh(ii,2)
           WRITE(NUT,105)tt,qh(ii,2)
```

```
      ELSE
      DO 25 k=1,nstep1
         IF(ii.eq.k*nWRITE1) THEN
            WRITE(NUT,106)tt,qh(ii,2)
         END IF
25    CONTINUE
      END IF
29    CONTINUE
      END IF !!ELSE ??
c---WRITE 0 entry after last step
      tEND1=qh(nhyd,1)*3600.d0
!      WRITE(12,106)tEND1,qh(nhyd,2)
!      WRITE(12,107)tEND1+300.d0,0.d0
     WRITE(NUT,106)tEND1,qh(nhyd,2)
     WRITE(NUT,107)tEND1+300.d0,0.d0
C---------------------------------------------------------------
C OUTPUT VFSMOD rainfall hyetograph: *.irn
C---------------------------------------------------------------
      nstep2=100
      IF(nhyet.le.nstep2)THEN
         WRITE(NUT,201)nhyet+1,rfix
         DO 31 ii=1,nhyet-1
            tt=rti(ii)*3600.d0
            IF (ii.eq.1) THEN
                WRITE(NUT,203) tt,rfi(ii)
            ELSE
               WRITE(NUT,204) tt,rfi(ii)
            END IF
31      CONTINUE
       ELSE
       nWRITE2=nhyet/nstep2+1
        WRITE(NUT,201)nhyet/nWRITE2+2,rfix
       DO 33 ii=1,nhyet-1
           tt=rti(ii)*3600.d0
           IF (ii.eq.1) THEN
!             WRITE(14,203) tt,rfi(ii)
              WRITE(NUT,203) tt,rfi(ii)
            ELSE
              DO 32 k=1,nstep2
                 IF(ii.eq.k*nWRITE2) THEN
!                WRITE(14,204) tt,rfi(ii)
                 WRITE(NUT,204) tt,rfi(ii)
C                WRITE(*,'(2i4,2e12.5)')ii,k,tt,rfi(ii)
C                WRITE(NUT,102)
           END IF
32      CONTINUE
      END IF
```

```
33    CONTINUE
      END IF
c---WRITE 0 entry after last step
      tEND2=rti(nhyet)*3600.d0
!      WRITE(14,204)tEND2,rfi(nhyet)
!      WRITE(14,205)tEND2+300.d0,0.d0
      WRITE(NUT,204)tEND2,rfi(nhyet)
      WRITE(NUT,205)tEND2+300.d0,0.d0
C-----------------------------------------------------------------
C  OUTPUT MESSAGE AT END OF PROGRAM
C-----------------------------------------------------------------
     !WRITE(*,*)
     !WRITE(*,*)'...FINISHED...','UH v3.0.1 2/2012'
     !WRITE(*,*)
      WRITE(NUT,*)
      WRITE(NUT,*)'...FINISHED...','UH v3.0.1 2/2012'
      WRITE(NUT,*)
C-----------------------------------------------------------------
C  OUTPUT - FORMAT
C-----------------------------------------------------------------
101  FORMAT (2x,i4,2x,f8.1,2x,f11.4,2x,f7.4,8x,
     C   'Npart, Coarse, Ci(g/cm3), Por')
102  FORMAT(2x,f10.7,2x,f7.1,21x,'Dp(cm), SG(g/cm3)')
103  FORMAT(2x,f7.1,2x,f7.1,21x,'Swidth(m), Slength(m)')
104  FORMAT(2x,i4,2x,e12.5,19x,'nbcroff, bcropeak (m3/s)')
105  FORMAT(2x,e12.5,2x,e12.5,10x,' Time(s), ro(m3/s)')
106  FORMAT(2x,e12.5,2x,e12.5)
107  FORMAT(2x,e12.5,2x,e12.5,/,30('-'))
160  FORMAT(72('-'))
201  FORMAT(i4,2x,e12.5,20x,' NRAIN, RPEAK(m/s)')
203  FORMAT(2x,e12.5,3x,e12.5,10x,'Time(s), Rainfall Rate
(m/s)')
204  FORMAT(2x,e12.5,3x,e12.5)
205  FORMAT(2x,e12.5,3x,e12.5,/,30('-'))
250  FORMAT('Time to Ponding=',f8.3,' hr')
260  FORMAT('Duration of Rainfall Excess=',f8.3,' hr')
270  FORMAT('Time to Peak After Shifting=',f8.3,' hr')
280  FORMAT('Time Correction to Match Hyetograph=',f8.3,' hr')

     RETURN
     END SUBROUTINE vfsout
```

LIST OF REFERENCES

Algoazany, A. S., P. K. Kalita, G. F. Czapar, and J. K. Mitchell. 2007. "Phosphorus Transport through Subsurface Drainage and Surface Runoff from a Flat Watershed in East Central Illinois, USA." *Journal of Environmental Quality* 36 (3): 681–93.

Arnold, J. G., and N. Fohrer. 2005. "SWAT2000: Current Capabilities and Research Opportunities in Applied Watershed Modelling." *Hydrological Processes* 19 (3): 563–72. https://doi.org/10.1002/hyp.5611.

Assefa, Yared, P. V. Vara Prasad, C. Foster, Y. Wright, S. Young, P. Bradley, M. Stamm, and I. A. Ciampitti. 2018. "Major Management Factors Determining Spring and Winter Canola Yield in North America." *Crop Science* 58 (1): 1–16.

Ayugi, B., G. Tan, W. Ullah, R. Boiyo, and V.Ongoma. 2019. "Inter-Comparison of Remotely Sensed Precipitation Datasets over Kenya during 1998–2016." *Atmospheric Research* 225 (September): 96–109.

Bagarello, Vincenzo, Emanuele Barca, Mirko Castellini, Massimo Iovino, Renato Morbidelli, Carla Saltalippi, and Alessia Flammini. 2021. "A Plot-Scale Uncertainty Analysis of Saturated Hydraulic Conductivity of a Clay Soil." *Journal of Hydrology* 596 (May): 125694. https://doi.org/10.1016/j.jhydrol.2020.125694.

Baker, C. J., and K.E. Saxton. Food and Agriculture Organization of the United Nations, eds. 2007. *No-Tillage Seeding in Conservation Agriculture*. 2nd ed. Wallingford, UK ; Cambridge, MA: Published jointly by Food and Agriculture Organization of the United Nations and Cabi Pub.

Baker, John M., Timothy J. Griffis, and Tyson E. Ochsner. 2012. "Coupling Landscape Water Storage and Supplemental Irrigation to Increase Productivity and Improve Environmental Stewardship in the U.S. Midwest." *Water Resources Research* 48 (5). https://doi.org/10.1029/2011WR011780.

Baldwin, E., C. Washington-Ottombre, J. Dell'Angelo, D. Cole, and T.Evans. 2016. "Polycentric Governance and Irrigation Reform in Kenya." *Governance* 29 (2): 207–25.

Barthold, Frauke K., and Ross A. Woods. 2015. "Stormflow Generation: A Meta-Analysis of Field Evidence from Small, Forested Catchments." *Water Resources Research* 51 (5): 3730–53. https://doi.org/10.1002/2014WR016221.

Basso, S., and G. Botter. 2012. "Streamflow Variability and Optimal Capacity of Run-of-River Hydropower Plants." *Water Resources Research* 48 (10). https://doi.org/10.1029/2012WR012017.

Batchelor, Charles, V. Ratna Reddy, Conor Linstead, Murli Dhar, Sumit Roy, and Rebecca May. 2014. "Do Water-Saving Technologies Improve Environmental Flows?" *Journal of Hydrology*, Creating Partnerships Between Hydrology and Social Science: A Priority for Progress, 518 (October): 140–49. https://doi.org/10.1016/j.jhydrol.2013.11.063.

Baumhardt, R. L., R. C. Schwartz, O. R. Jones (retired), B. R. Scanlon, R. C. Reedy, and G. W. Marek. 2017. “Long-Term Conventional and No-Tillage Effects on Field Hydrology and Yields of a Dryland Crop Rotation.” *Soil Science Society of America Journal* 81 (1): 200–209.

Baveye, P. C., D. Rangel, A.R. Jacobson, M. Laba, C. Darnault, W. Otten, R. Radulovich, and F.A. O. Camargo. 2011. “From Dust Bowl to Dust Bowl: Soils Are Still Very Much a Frontier of Science.” *Soil Science Society of America Journal* 75 (6): 2037–48. https://doi.org/10.2136/sssaj2011.0145.

Bergström, Sten. 1991. “Principles and Confidence in Hydrological Modelling.” *Hydrology Research* 22 (2): 123–36. https://doi.org/10.2166/nh.1991.0009.

Beven, Keith. 1989. “Changing Ideas in Hydrology — The Case of Physically-Based Models.” *Journal of Hydrology* 105 (1): 157–72. https://doi.org/10.1016/0022-1694(89)90101-7.

2006. “On Undermining the Science?” *Hydrological Processes* 20 (14): 3141–46. https://doi.org/10.1002/hyp.6396.

Birkel, Christian, Doerthe Tetzlaff, Sarah M. Dunn, and Chris Soulsby. 2011. “Using Time Domain and Geographic Source Tracers to Conceptualize Streamflow Generation Processes in Lumped Rainfall-Runoff Models.” *Water Resources Research* 47 (2). https://doi.org/10.1029/2010WR009547.

Blanco-Canqui, H., and S.J. Ruis. 2018. “No-Tillage and Soil Physical Environment.” *Geoderma* 326 (September): 164–200.

Bosch, D.D., C.C. Truman, T. L. Potter, L.T. West, T. C. Strickland, and R.K. Hubbard. 2012. “Tillage and Slope Position Impact on Field-Scale Hydrologic Processes in the South Atlantic Coastal Plain.” *Agricultural Water Management* 111 (August): 40–52.

Bowmer, K. H. 2011. “Water Resource Protection in Australia: Links between Land Use and River Health with a Focus on Stubble Farming Systems.” *Journal of Hydrology* 403 (1): 176–85.

Brakenridge, R., and E. Anderson. 2006. “MODIS-Based Flood Detection, Mapping, and Measurement: The Potential For Operational Hydrological Applications”In *Transboundary Floods: Reducing Risks Through Flood Management*, edited by Jiri Marsalek, Gheorghe Stancalie, and Gabor Balint, 1–12. Nato Science Series: IV: Earth and Environmental Sciences. Dordrecht: Springer Netherlands. https://doi.org/10.1007/1-4020-4902-1_1.

Brouder, S.M., and H.Gomez-Macpherson. 2014. “The Impact of Conservation Agriculture on Smallholder Agricultural Yields: A Scoping Review of the Evidence.” *Agriculture, Ecosystems & Environment,* Evaluating conservation agriculture for small-scale farmers in Sub-Saharan Africa and South Asia, 187 (April): 11–32.

Burt, T. P., and J. J. McDonnell. 2015. "Whither Field Hydrology? The Need for Discovery Science and Outrageous Hydrological Hypotheses." *Water Resources Research* 51 (8): 5919–28. https://doi.org/10.1002/2014WR016839.

Campo-Bescós, M.A., R. Muñoz-Carpena, D.A. Kaplan, J. Southworth, L. Zhu, and P.R. Waylen. 2013. "Beyond Precipitation: Physiographic Gradients Dictate the Relative Importance of Environmental Drivers on Savanna Vegetation." *PLOS ONE* 8 (8): e72348.

Churcher, C. S. 1993. "Equus Grevyi." *Mammalian Species*, no. 453: 1–9.

Clark, Martyn P., Robert L. Wilby, Ethan D. Gutmann, Julie A. Vano, Subhrendu Gangopadhyay, Andrew W. Wood, Hayley J. Fowler, Christel Prudhomme, Jeffrey R. Arnold, and Levi D. Brekke. 2016. "Characterizing Uncertainty of the Hydrologic Impacts of Climate Change." *Current Climate Change Reports* 2 (2): 55–64. https://doi.org/10.1007/s40641-016-0034-x.

Coppock, D.L., J.E. Ellis, and D.M. Swift. 1988. "Seasonal Patterns of Activity, Travel and Water Intake of Livestock in South Turkana, Kenya." *Journal of Arid Environments* 14: 319–31.

Cui, G., X. Wang, C. Li, Y. Li, S. Yan, and Z. Yang. 2018. "Water Use Efficiency and TN/TP Concentrations as Indicators for Watershed Land-Use Management: A Case Study in Miyun District, North China." *Ecological Indicators* 92 (September): 239–53.

Dariane, Alireza B., M. M. Javadianzadeh, and L. Douglas James. 2016. "Developing an Efficient Auto-Calibration Algorithm for HEC-HMS Program." *Water Resources Management* 30 (6): 1923–37. https://doi.org/10.1007/s11269-016-1260-7.

Derpsch, R. 2008. "No-Tillage and Conservation Agriculture: A Progress Report," 25.

Didoné, E.J., J.P. Gomes Minella, J. Miguel Reichert, G. Henrique Merten, L. Dalbianco, C.A. Peixoto de Barrros, and R. Ramon. 2014. "Impact of No-Tillage Agricultural Systems on Sediment Yield in Two Large Catchments in Southern Brazil." Journal of Soils and Sediments, 14, 1287–1297.

Dile, Y.T., L. Karlberg, M. Temesgen, and J. Rockström. 2013. "The Role of Water Harvesting to Achieve Sustainable Agricultural Intensification and Resilience against Water Related Shocks in Sub-Saharan Africa." *Agriculture, Ecosystems & Environment* 181 (December): 69–79.

Eckert, Sandra, Boniface Kiteme, Evanson Njuguna, and Julie Gwendolin Zaehringer. 2017. "Agricultural Expansion and Intensification in the Foothills of Mount Kenya: A Landscape Perspective." *Remote Sensing* 9 (8): 784. https://doi.org/10.3390/rs9080784.

Emmanuel, A.T. 2012. “Modernisation of Smallholder Agriculture and Policy Making in Uganda,” Building an Effective Advocacy Movement for Sustainable and Equitable Agricultural Development in Africa, , 33.

ESA. 2017. Land Cover CCI Product User Guide Version 2. Tech. Rep. Available at: maps.elie.ucl.ac.be/CCI/viewer/download/ESACCI-LC-Ph2-PUGv2_2.0.pdf

ESA-CCI. 2017. CCI Land Cover – S2 Prototype Land Cover 20m Map of Africa 2016. Available at: https://2016africalandcover20m.esrin.esa.int/download.php

Endale, D. M., D. D. Bosch, T. L. Potter, and T. C. Strickland. 2014. “Sediment Loss and Runoff from Cropland in a Southeast Atlantic Coastal Plain Landscape.” *Transactions of the ASABE* 57 (6): 1611–26.

Fan, Z., Q. Chai, G. Huang, A. Yu, P. Huang, C. Yang, Z. Tao, and H. Liu. 2013. “Yield and Water Consumption Characteristics of Wheat/Maize Intercropping with Reduced Tillage in an Oasis Region.” European Journal of Agronomy 45 (February): 52–58.

Farmer, William H., and Richard M. Vogel. 2016. “On the Deterministic and Stochastic Use of Hydrologic Models.” *Water Resources Research* 52 (7): 5619–33. https://doi.org/10.1002/2016WR019129.

Fleming, Matt, and Vincent Neary. 2004. “Continuous Hydrologic Modeling Study with the Hydrologic Modeling System.” *Journal of Hydrologic Engineering* 9 (3): 175–83. https://doi.org/10.1061/(ASCE)1084-0699(2004)9:3(175).

Fodor, Nándor, Renáta Sándor, Tomas Orfanus, Lubomir Lichner, and Kálmán Rajkai. 2011. “Evaluation Method Dependency of Measured Saturated Hydraulic Conductivity.” *Geoderma* 165 (1): 60–68. https://doi.org/10.1016/j.geoderma.2011.07.004.

Frame, Jonathan M., Frederik Kratzert, Hoshin V. Gupta, Paul Ullrich, and Grey S. Nearing. 2023. “On Strictly Enforced Mass Conservation Constraints for Modelling the Rainfall-Runoff Process.” *Hydrological Processes* 37 (3): e14847. https://doi.org/10.1002/hyp.14847.

Freeze, R. Allan, and R. L. Harlan. 1969. “Blueprint for a Physically-Based, Digitally-Simulated Hydrologic Response Model.” *Journal of Hydrology* 9 (3): 237–58. https://doi.org/10.1016/0022-1694(69)90020-1.

Fuentes, Juan P., Markus Flury, and David F. Bezdicek. 2004. “Hydraulic Properties in a Silt Loam Soil under Natural Prairie, Conventional Till, and No-Till.” *Soil Science Society of America Journal* 68 (5): 1679–88. https://doi.org/10.2136/sssaj2004.1679.

Funk, C., Peterson, P., Landsfeld, M. et al. 2015.The climate hazards infrared precipitation with stations—a new environmental record for monitoring extremes. Sci Data 2, 150066.

Food and Agriculture Organization (FAO). 1998. FAO-56 Crop Evapotranspiration: Guidelines for Computing Crop Requirements. http:// www.fao.org/docrep/X0490E/X0490E00.htm, accessed November 10, 2022.

FAO. 2020a. “AQUASTAT - FAO’s Global Information System on Water and Agriculture.” 2020. http://www.fao.org/aquastat/en/.

FAO. 2020b. “Conservation Agriculture | Food and Agriculture Organization of the United Nations.” 2020. http://www.fao.org/conservation-agriculture/en/.

Garnett, T., M. C. Appleby, A. Balmford, I. J. Bateman, T. G. Benton, P. Bloomer, B. Burlingame, et al. 2013. “Sustainable Intensification in Agriculture: Premises and Policies.” *Science* 341 (6141): 33–34. https://doi.org/10.1126/science.1234485.

Giller, K.E., E. Witter, M. Corbeels, and P. Tittonell. 2009. “Conservation Agriculture and Smallholder Farming in Africa: The Heretics’ View.” *Field Crops Research* 114 (1): 23–34.

Good, S.P. and K.K. Caylor. 2011. “Climatological Determinants of Woody Cover in Africa.” *Proceedings of the National Academy of Sciences* 108 (12): 4902–7.

Gower, D.B, J. Dell’Angelo, P.F. McCord, K.K. Caylor, and T.P. Evans. 2016. “Modeling Ecohydrological Dynamics of Smallholder Strategies for Food Production in Dryland Agricultural Systems.” Environmental Research Letters 11 (11): 115005.

Grafton, R. Q., J. Williams, C. J. Perry, F. Molle, C. Ringler, P. Steduto, B. Udall, et al. 2018. “The Paradox of Irrigation Efficiency.” Science 361 (6404): 748–50.

Green, W.H. and G. A. Ampt. 1911. “Studies on Soil Phyics.” The Journal of Agricultural Science 4 (1). Cambridge University Press: 1–24. doi:10.1017/S0021859600001441.

Haberlandt, U. 2010. “From Hydrological Modelling to Decision Support.” *Advances in Geosciences* 27 (August): 11–19. https://doi.org/10.5194/adgeo-27-11-2010.

Haan, C.T., H.P. Johnson and D.L. Brakenslek (Ed.). 1982. Hydrologic Modeling of Small Watersheds. ASAE Monograph No.5. American Society ofAgricultural Engineers (ASABE), St. Joseph. Michigan

Hengl T., Heuvelink GBM, Kempen B, Leenaars JGB, Walsh MG, Shepherd KD, et al. 2015. Mapping Soil Properties of Africa at 250 m Resolution: Random Forests Significantly Improve Current Predictions. PLoS ONE 10(6): e0125814.

Hjelmfelt, Allen T. 1991. “Investigation of Curve Number Procedure.” *Journal of Hydraulic Engineering* 117 (6): 725–37. https://doi.org/10.1061/(ASCE)0733-9429(1991)117:6(725).

Hobbs, P.R, K. Sayre, and R. Gupta. 2008. “The Role of Conservation Agriculture in Sustainable Agriculture.” *Philosophical Transactions of the Royal Society B: Biological Sciences* 363 (1491): 543–55.

Hulley, G.. 2021a. *MODIS/Aqua Land Surface Temperature/3-Band Emissivity Daily L3 Global 1km SIN Grid Day V061*. distributed by NASA EOSDIS Land Processes DAAC, https://doi.org/10.5067/MODIS/MYD21A1D.061. Accessed 2023-06-27.

Hulley, G.. 2021b. *MODIS/Aqua Land Surface Temperature/3-Band Emissivity Daily L3 Global 1km SIN Grid Night V061*. distributed by NASA EOSDIS Land Processes DAAC, https://doi.org/10.5067/MODIS/MYD21A1N.061. Accessed 2023-06-27.

Hromadka, T.V., T.J. Durbin, and J.J. Devries. 1985. *Computer Methods in Water Resources*. USA: Lighthouse Publications.

Huffaker, R. 2008. “Conservation Potential of Agricultural Water Conservation Subsidies.” *Water Resources Research* 44 (7). h

Jägermeyr, J., D. Gerten, J. Heinke, S. Schaphoff, M. Kummu, and W. Lucht. 2015. “Water Savings Potentials of Irrigation Systems: Dynamic Global Simulation.” *Hydrology and Earth System Sciences Discussions* 12 (4): 3593–3644.

Jägermeyr, J, D Gerten, S Schaphoff, J Heinke, W Lucht, and J Rockström. 2016. “Integrated Crop Water Management Might Sustainably Halve the Global Food Gap.” *Environmental Research Letters* 11 (2): 025002.

Johnstone, Don. n.d. “Elements of Applied Hydrology.” *(No Title)*. Accessed August 14, 2023. https://cir.nii.ac.jp/crid/1130282273363995648.

Kannan, N., S. M. White, F. Worrall, and M. J. Whelan. 2007. “Sensitivity Analysis and Identification of the Best Evapotranspiration and Runoff Options for Hydrological Modelling in SWAT-2000.” *Journal of Hydrology* 332 (3): 456–66. https://doi.org/10.1016/j.jhydrol.2006.08.001.

Kassam, A., R. Derpsch, and T. Friedrich. 2014. “Global Achievements in Soil and Water Conservation: The Case of Conservation Agriculture.” *International Soil and Water Conservation Research* 2 (1): 5–13.

Kassam, A., T. Friedrich, and R. Derpsch. 2010. “Conservation Agriculture in the 21st Century: A Paradigm of Sustainable Agriculture,” Europiean Congress on Conservation Agriculture, 46.

Kiptala, J. K., Mohamed, Y., Mul, M. L., and Van der Zaag, P. (2013), Mapping evapotranspiration trends using MODIS and SEBAL model in a data scarce and heterogeneous landscape in Eastern Africa, Water Resour. Res., 49, 8495– 8510.

Kirchner, James W. 2006. “Getting the Right Answers for the Right Reasons: Linking Measurements, Analyses, and Models to Advance the Science of Hydrology.” *Water Resources Research* 42 (3). https://doi.org/10.1029/2005WR004362.

Laikipia, County Government of. 2020a. “INVESTING IN LAIKIPIA.” County Government of Laikipia. 2020. https://laikipia.go.ke/270/investing-in-laikipia/.

Laikipia, County Government of. 2020b. “Successes: Promotion of Drip Irrigation.” County Government of Laikipia. 2020. https://laikipia.go.ke/114/successes-promotion-drip-irrigation/.

Laikipia Wildlife Forum, KENWEB & NMK. 2013. “A Conservation Strategy for Laikipia County (2014-2018).” Laikipia Wildlife Forum.

Lal, R. 2001. “Soil Degradation by Erosion.” *Land Degradation & Development* 12 (6): 519–39.

Lanari, N., R. Schuler, T. Kohler, and H. Liniger. 2018. “The Impact of Commercial Horticulture on River Water Resources in the Upper Ewaso Ng’iro River Basin, Kenya.” *Mountain Research and De*velopment 38 (2): 114–24. https://doi.org/10.1659/MRD-JOURNAL-D-16-00135.

Leong, C., and Y. Yokoo. 2021. “A Step toward Global-Scale Applicability and Transferability of Flow Duration Curve Studies: A Flow Duration Curve Review (2000–2020).” *Journal of Hydrology* 603 (December): 126984. https://doi.org/10.1016/j.jhydrol.2021.126984.

Livingston, G., S. Schonberger, and S.D. Session. 2011. *Sub-Saharan Africa: The State of Smallholders in Agriculture*. IFAD Conference on New Directions for Smallholder Agriculture.

Luquin, E., M. A. Campo-Bescós, R. Muñoz-Carpena, R.L. Bingner, R.M. Cruse, H.G. Momm, R.R. Wells, and J. Casalí. 2021. “Model Prediction Capacity of Ephemeral Gully Temporal Evolution in Conservation Tillage Systems.” *Earth Surface Processes and Landforms*

Macharia, J.M., F.K. Ngetich, and C.A. Shisanya. 2020. “Comparison of Satellite Remote Sensing Derived Precipitation Estimates and Observed Data in Kenya.” *Agricultural and Forest Meteorology* 284 (April): 107875.

Makurira, H., H. H. G. Savenije, S. Uhlenbrook, J. Rockström, and A. Senzanje. 2007. “Towards a Better Understanding of Water Partitioning Processes for Improved Smallholder Rainfed Agricultural Systems: A Case Study of Makanya Catchment, Tanzania.” *Physics and Chemistry of the Earth, Parts A/B/C*, Mainstreaming Integrated Water Resources Management in the Development Process, 32 (15): 1082–89.

Moriasi, D.N., J. G. Arnold, M. W. Van Liew, R. L. Bingner, R. D. Harmel, and T. L. Veith. 2007. “Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulations.” *Transactions of the ASABE* 50 (3): 885–900.

Mueller, N.D., J.S. Gerber, M. Johnston, D. K. Ray, N. Ramankutty, and J.A. Foley. 2012. “Closing Yield Gaps through Nutrient and Water Management.” *Nature* 490 (7419): 254–57.

Muller, S., R. Muñoz-Carpena, G. Kiker. 2011. Model Relevance: Frameworks for Exploring the Complexity-Sensitivity-Uncertainty Trilemma. In: I. Linkov and T.S.S. Bridges (eds.).Climate: Global Change and Local Adaptation. NATO Science for Peace and Security Series C: Environmental Security. pp. 35-65. Springer:Boston doi:10.1007/978-94-007-1770-1_4.

Muñoz-Carpena, R., J.E. Parsons, and J.W. Gilliam. 1999. Modeling hydrology and sediment transport in vegetative filter strips. *Journal of Hydrology* 214(1-4):111-129. doi: 10.1016/S0022-1694(98)00272-8

Muñoz-Carpena, R., and J. E. Parsons. 2004. “A DESIGN PROCEDURE FOR VEGETATIVE FILTER STRIPS USING VFSMOD-W.” *Transactions of the ASAE* 47 (6): 1933–41.

Muñoz-Carpena, R., G. Vellidis, A. Shirmohammadi and W.W. Wallender. 2006. Evaluation of Modeling Tools for TMDL Development and Implementation. Trans. of ASABE 49(4):961-965.

Muñoz-Carpena, Rafael, A Carmona-Cabrero, Z Yu, G Fox, and O Batelaan. 2023. “Convergence of Mechanistic Modeling and Artificial Intelligence in Hydrologic Science and Engineering.” *PLOS Water* 2 (8): e0000059. https://doi.org/10.1371/journal.pwat.0000059.

Mupangwa, W., S. Twomlow, S. Walker, and L. Hove. 2007. “Effect of Minimum Tillage and Mulching on Maize (Zea Mays L.) Yield and Water Content of Clayey and Sandy Soils.” *Physics and Chemistry of the Earth, Parts A/B/C,* Mainstreaming Integrated Water Resources Management in the Development Process, 32 (15): 1127–34.

Muriithi, Faith K. 2016. “Land Use and Land Cover (LULC) Changes in Semi-Arid Sub-Watersheds of Laikipia and Athi River Basins, Kenya, as Influenced by Expanding Intensive Commercial Horticulture.” *Remote Sensing Applications: Society and Environment* 3 (May): 73–88. https://doi.org/10.1016/j.rsase.2016.01.002.

Nakawuka, P., S. Langan, P. Schmitter, and J.Barron. 2018. “A Review of Trends, Constraints and Opportunities of Smallholder Irrigation in East Africa.” *Global Food Security* 17 (June): 196–212.

Ndomba, Preksedis, Felix Mtalo, and Aanund Killingtveit. 2008. “SWAT Model Application in a Data Scarce Tropical Complex Catchment in Tanzania.” *Physics and Chemistry of the Earth, Parts A/B/C,* Integrated Water Resources Management - From Concept to Practice, 33 (8): 626–32. https://doi.org/10.1016/j.pce.2008.06.013.

Ngigi, S.N., Hubert H. G. Savenije, and F.N. Gichuki. 2008. “Hydrological Impacts of Flood Storage and Management on Irrigation Water Abstraction in Upper Ewaso Ng’iro River Basin, Kenya.” *Water Resources Management* 22 (12): 1859–79.

Nyagumbo, I., W. Mupangwa, L.Chipindu, L. Rusinamhodzi, and P. Craufurd. 2020. “A Regional Synthesis of Seven-Year Maize Yield Responses to Conservation Agriculture Technologies in Eastern and Southern Africa.” *Agriculture, Ecosystems & Environment* 295 (June): 106898.

O’Geen, Anthony, Mike Walkinshaw, and Dylan Beaudette. 2017. “SoilWeb: A Multifaceted Interface to Soil Survey Information.” *Soil Science Society of America Journal* 81 (4): 853–62. https://doi.org/10.2136/sssaj2016.11.0386n.

Ogutu, Joseph O., Hans-Peter Piepho, Mohamed Y. Said, Gordon O. Ojwang, Lucy W. Njino, Shem C. Kifugo, and Patrick W. Wargute. 2016. “Extreme Wildlife Declines and Concurrent Increase in Livestock Numbers in Kenya: What Are the Causes?” *PLOS ONE* 11 (9): e0163249. https://doi.org/10.1371/journal.pone.0163249.

Peschel, Joshua M., Patricia K. Haan, and Ronald E. Lacey. 2006. “Influences of Soil Dataset Resolution on Hydrologic Modeling1.” *JAWRA Journal of the American Water Resources Association* 42 (5): 1371–89. https://doi.org/10.1111/j.1752-1688.2006.tb05307.x.

Pittelkow, C.M., B.A. Linquist, M.E. Lundy, X. Liang, K. Jan van Groenigen, J. Lee, N. van Gestel, J. Six, R.T. Venterea, and C. van Kessel. 2015. “When Does No-till Yield More? A Global Meta-Analysis.” Field Crops Research 183 (November): 156–68. https://doi.org/10.1016/j.fcr.2015.07.020.

Pool, S., F. Francés, A. Garcia-Prats, M. Pulido-Velazquez, C. Sanchis-Ibor, M. Schirmer, H. Yang, and J. Jiménez-Martínez. 2021. “From Flood to Drip Irrigation Under Climate Change: Impacts on Evapotranspiration and Groundwater Recharge in the Mediterranean Region of Valencia (Spain).” *Earth’s Future* 9 (5): e2020EF001859. https://doi.org/10.1029/2020EF001859.

Pretty, J., C. Toulmin, and S.Williams. 2011. “Sustainable Intensification in African Agriculture.” International Journal of Agricultural Sustainability 9 (1): 5–24.

Reuter H.I, A. Nelson, A. Jarvis. 2007. An evaluation of void filling interpolation methods for SRTM data, International Journal of Geographic Information Science, 21:9, 983-1008

Reynolds, W. D., B. T. Bowman, R. R. Brunke, C. F. Drury, and C. S. Tan. 2000. “Comparison of Tension Infiltrometer, Pressure Infiltrometer, and Soil Core Estimates of Saturated Hydraulic Conductivity.” *Soil Science Society of America Journal* 64 (2): 478–84. https://doi.org/10.2136/sssaj2000.642478x.

Ricciardi, V., N. Ramankutty, Z. Mehrabi, L. Jarvis, and B. Chookolingo. 2018. “How Much of the World’s Food Do Smallholders Produce?” *Global Food Security* 17 (June): 64–72.

Ritter, A., and R. Muñoz-Carpena. 2013. “Performance Evaluation of Hydrological Models: Statistical Significance for Reducing Subjectivity in Goodness-of-Fit Assessments.” *Journal of Hydrology* 480 (February): 33–45.

Rockström, J., P. Kaumbutho, J. Mwalley, A. W. Nzabi, M. Temesgen, L. Mawenya, J. Barron, J. Mutua, and S. Damgaard-Larsen. 2009. “Conservation Farming Strategies in East and Southern Africa: Yields and Rain Water Productivity from on-Farm Action Research.” *Soil and Tillage Research* 103 (1): 23–32.

Rockström, J., and L .Karlberg. 2010. “The Quadruple Squeeze: Defining the Safe Operating Space for Freshwater Use to Achieve a Triply Green Revolution in the Anthropocene.” *Ambio* 39 (3): 257–65.

Rockström, J., J. Williams, G. Daily, A. Noble, N. Matthews, L. Gordon, H. Wetterstrand, et al. 2017. “Sustainable Intensification of Agriculture for Human Prosperity and Global Sustainability.” *Ambio* 46 (1): 4–17.

Salami, A., A.B. Kamara, and Z. Brixiova. 2010. “Smallholder Agriculture in East Africa: Trends, Constraints and Opportunities,” Working Papers Series N 105 African Development Bank, Tunis, Tunisia, , 52.

Salimi, E. Taghvaye, A. Nohegar, A. Malekian, M. Hoseini, and A. Holisaz. 2017. “Estimating Time of Concentration in Large Watersheds.” *Paddy and Water Environment* 15 (1): 123–32. https://doi.org/10.1007/s10333-016-0534-2.

Samberg, L.H., J.S. Gerber, N. Ramankutty, M. Herrero, and P.C. West. 2016. “Subnational Distribution of Average Farm Size and Smallholder Contributions to Global Food Production.” *Environmental Research Letters* 11 (12): 124010.

Sankaran, M., N.P. Hanan, R.J. Scholes, J. Ratnam, D.J. Augustine, B.S. Cade, J. Gignoux, et al. 2005. “Determinants of Woody Cover in African Savannas.” *Nature* 438 (7069): 846–49.

Scholes, R. J., and S. R. Archer. 1997. “Tree-Grass Interactions in Savannas.” *Annual Review of Ecology and Systematics* 28 (1): 517–44.

Schulla, Jörg, and Karsten Jasper. n.d. “Model Description WaSiM-ETH.”

Schultz, B., C. D. Thatte, and V. K. Labhsetwar. 2005. “Irrigation and Drainage. Main Contributors to Global Food Production.” *Irrigation and Drainage* 54 (3): 263–78.

Schulze, R.E., 1995. “Hydrology and agrohydrology: A text to accompany the ACRU 3.00 agrohydrological modelling system.” Water Research Commission.

Seckler, D.W. 1996. The New Era of Water Resources Management: From “Dry” to “Wet” Water Savings. IWMI.

Shirmohammadi, A., I. Chaubey, R.D. Harmel, D.D. Bosch, R. Muñoz-Carpena, C. Dharmasri, A. Sexton, M. Arabi, M.L. Wolfe, J. Frankenberger, C. Graff and T.M. Sohrabi. 2006. Uncertainty in TMDL Models. Trans. of ASABE 49(4):1033-1049. doi: 10.13031/2013.21741

Sithole, N.J., L. S. Magwaza, and G.R. Thibaud. 2019. "Long-Term Impact of No-till Conservation Agriculture and N-Fertilizer on Soil Aggregate Stability, Infiltration and Distribution of C in Different Size Fractions." *Soil and Tillage Research* 190 (July): 147–56.

Skaggs, R.W., Khaheel, R., 1982. Infiltration, Ch. 4, In: Hydrologic modeling of small watersheds. C.T. Haan, H.P. Johnson, D.L. Brakensiek, Eds., St. Joseph, MI, ASAE. pp. 121–168.

Slack, J. R., and Jurate Maciunas Landwehr. 1992. "Hydro-Climatic Data Network (HCDN); a U.S. Geological Survey Streamflow Data Set for the United States for the Study of Climate Variations, 1874-1988." USGS Numbered Series 92–129. *Hydro-Climatic Data Network (HCDN); a U.S. Geological Survey Streamflow Data Set for the United States for the Study of Climate Variations, 1874-1988*. Vol. 92–129. Open-File Report. U.S. Geological Survey ; Copies of this report can be purchased from USGS Books and Open-File Reports Section,. https://doi.org/10.3133/ofr92129.

Smakhtin, V. U., and N. Eriyagama. 2008. "Developing a Software Package for Global Desktop Assessment of Environmental Flows." *Environmental Modelling & Software* 23 (12): 1396–1406. https://doi.org/10.1016/j.envsoft.2008.04.002.

Smit, I.P. J., C.C. Grant, and B.J. Devereux. 2007. "Do Artificial Waterholes Influence the Way Herbivores Use the Landscape? Herbivore Distribution Patterns around Rivers and Artificial Surface Water Sources in a Large African Savanna Park." *Biological Conservation*, Special section: Coastal Sandplains, 136 (1): 85–99.

Smith, M., G. Muñoz, and J. Sanz Alvarez. 2014. "Irrigation Techniques for Small-Scale Farmers: Key Practices for DRR Implementers," A Field Guide for Disaster Risk Reduction in Southern Africa: Key Practices for DRR Implementers. FAO., , 52.

Southworth, J., E. Bunting, L. Zhu, S.J. Ryan, H.V. Herrero, P.Waylen, R. Muñoz-Carpena, M.A. Campo-Bescós, and D. Kaplan. 2018. "Using a Coupled Dynamic Factor – Random Forest Analysis (DFRFA) to Reveal Drivers of Spatiotemporal Heterogeneity in the Semi-Arid Regions of Southern Africa." *PLOS ONE* 13 (12): e0208400.

Steward, R., A.J. Dougill, C. Thierfelder, C.M. Pittelkow, L.C. Stringer, M. Kudzala, and G. E. Shackelford. 2018. "The Adaptive Capacity of Maize-Based Conservation Agriculture Systems to Climate Stress in Tropical and Subtropical Environments: A Meta-Regression of Yields." *Agriculture, Ecosystems & Environment* 251 (January): 194–202.

Sudheer, K. P., I. Chaubey, V. Garg, and Kati W. Migliaccio. 2007. "Impact of Time-Scale of the Calibration Objective Function on the Performance of Watershed Models." *Hydrological Processes* 21 (25): 3409–19. https://doi.org/10.1002/hyp.6555.

Thierfelder, C., F. Baudron, P.Setimela, I. Nyagumbo, W. Mupangwa, B. Mhlanga, N. Lee, and B. Gérard. 2018. “Complementary Practices Supporting Conservation Agriculture in Southern Africa. A Review.” *Agronomy for Sustainable Development* 38 (2): 16.

Thierfelder, C., R. Matemba-Mutasa, and L. Rusinamhodzi. 2015. “Yield Response of Maize (Zea Mays L.) to Conservation Agriculture Cropping System in Southern Africa.” *Soil and Tillage Research* 146 (March): 230–42.

Tomer, M. D., and M. A. Locke. 2011. “The Challenge of Documenting Water Quality Benefits of Conservation Practices: A Review of USDA-ARS’s Conservation Effects Assessment Project Watershed Studies.” *Water Science and Technology* 64 (1): 300–310.

Tomer, M. D., D. W. Meek, and L. A. Kramer. 2005. “Agricultural Practices Influence Flow Regimes of Headwater Streams in Western Iowa.” *Journal of Environmental Quality* 34 (5): 1547–58.

Ullrich, A., and M. Volk. 2009. “Application of the Soil and Water Assessment Tool (SWAT) to Predict the Impact of Alternative Management Practices on Water Quality and Quantity.” *Agricultural Water Management* 96 (8): 1207–17.

USDA-NRCS. 1986. *Urban Hydrology for Small Watersheds*. 210−VI−TR−55, 2nd ed. Washington, D.C.: USDA-NRCS.

Valbuena, D., O. Erenstein, S. Homann-Kee Tui, T. Abdoulaye, L. Claessens, A.J. Duncan, B. Gérard. 2012. “Conservation Agriculture in Mixed Crop–Livestock Systems: Scoping Crop Residue Trade-Offs in Sub-Saharan Africa and South Asia.” *Field Crops Research*, Conservation Agriculture in Dry Areas, 132 (June): 175–84.

Van der Kooij, Saskia, Margreet Zwarteveen, Harm Boesveld, and Marcel Kuper. 2013. “The Efficiency of Drip Irrigation Unpacked.” *Agricultural Water Management* 123 (May): 103–10. https://doi.org/10.1016/j.agwat.2013.03.014.

Veith, T L, M W Van Liew, D D Bosch, and J G Arnold. 2010. “Parameter Sensitivity and Uncertainty in SWAT: A Comparison across Five USDA-ARS Watersheds.” *TRANSACTIONS OF THE ASABE* 53 (5): 1477–86.

Verhulst, N, B Govaerts, Els Verachtert, Antonio Castellanos-Navarrete, Monica Mezzalama, Wall C., Jozef Deckers, and K Sayre. 2010. “Conservation Agriculture, Improving Soil Quality for Sustainable Production Systems?” In *Advances in Soil Science: Food Security and Soil Quality*, 137–208.

Vigerstol, Kari L., and Juliann E. Aukema. 2011. “A Comparison of Tools for Modeling Freshwater Ecosystem Services.” *Journal of Environmental Management* 92 (10): 2403–9. https://doi.org/10.1016/j.jenvman.2011.06.040.

Voinov, Alexey, and Erica J. Brown Gaddis. 2008. “Lessons for Successful Participatory Watershed Modeling: A Perspective from Modeling Practitioners.” *Ecological Modelling*, Special Issue dedicated to the memory of Yuri Svirezhev, 216 (2): 197–207. https://doi.org/10.1016/j.ecolmodel.2008.03.010.

Wall, P. C. 2007. “Tailoring Conservation Agriculture to the Needs of Small Farmers in Developing Countries: An Analysis of Issues.” *Journal of Crop Improvement* 19 (1–2): 137–55.

Ward, F. A., and M. Pulido-Velazquez. 2008. “Water Conservation in Irrigation Can Increase Water Use.” *Proceedings of the National Academy of Sciences* 105 (47): 18215–20.

Wiesmann, U., F.N. Gichuki, B.P. Kiteme, and H. Liniger. 2000. “Mitigating Conflicts Over Scarce Water Resources in the Highland-Lowland System of Mount Kenya.” *Mountain Research and Development* 20 (1): 10–15.

Zanaga, D., R. Van De Kerchove, W. De Keersmaecker, N. Souverijns, C. Brockmann, R. Quast, J. Wevers, A. Grosu, A. Paccini, S. Vergnaud, O. Cartus, M. Santoro, S. Fritz, I. Georgieva, M. Lesiv, S. Carter, M. Herold, Linlin Li, N.E. Tsendbazar, F. Ramoino, O. Arino. 2021. ESA WorldCover 10 m 2020 v100.

Zanaga, D., R. Van De Kerchove, D. Daems, W. De Keersmaecker, C. Brockmann, G. Kirches, J. Wevers, O. Cartus, M. Santoro, S. Fritz, M. Lesiv, M. Herold, N.E. Tsendbazar, P. Xu, F. Ramoino, O. Arino. 2022. ESA WorldCover 10 m 2021 v200.

BIOGRAPHICAL SKETCH

Lory Willard obtained a bachelor's degree from Virginia Tech from the Department of Biological Systems Engineering in 2013 and a master's degree from the same department in 2014. During her master's degree, she conducted research on the pollutant removal and flow reduction efficiencies of a bioretention cell in Blacksburg, Virginia. From 2014 to 2016, she worked for an environmental consulting firm in Annapolis, Maryland, where she focused on field studies and design of stream and wetland mitigation projects, green stormwater infrastructure implementation, fish passage projects, most of which were for government agencies as part of the Chesapeake Bay TMDL program. From 2016 to 2018, she returned to her home state of North Carolina to work for the City of Raleigh Stormwater Management Division where she worked in the Water Quality Section managing projects to meet EPA NPDES and NC Neuse Rules permit requirements. She also managed the Raleigh Rainwater Rewards program and was able to support green stormwater initiatives throughout Raleigh with education and policy initiatives. In 2018, she received a departmental fellowship to pursue her Ph.D. at the University of Florida from the Agricultural and Biological Engineering Department. Her research focused on the hydrology of sustainable intensification practices in Laikipia, Kenya, where she was fortunate to spend several months in the field. While pursuing her Ph.D., she was a teaching assistant for the Land and Water Engineering course, manuscript screener for the high-tier journal "Journal of Hydrology: Regional Studies" and organizer of the biocomplexity engineering seminars.