MASTERS PLAN B REPORT

A WEB BASED APPLICATION PROTOTYPE RIVER ROUTING MODEL TO FACILITATE RESERVOIR RELEASE SCHEDULING

Submitted by

Patrick Wynn Noe

Department of Civil and Environmental Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Fall 2021

Master's Committee:

Advisor: Dr. Neil Grigg

Dr. Jeffrey Niemann

Dr. Sangmi Pallickara

Table of Contents

Table Of Figures

1. Introduction

Building models for systems in Water Resources Management has become central to providing engineering services in the discipline. Advances in computing technologies continually bring with it the ability to model more complex physical processes related to water resources, and brilliant models have been developed that help to answer important questions in policy and decision making. However, model complexity often leads to lack of model usability. An important aspect of model design that becomes increasingly challenging with increasing model complexity is maintaining an adequate level of usability of the model so that the end user feels both confident in the tool's capabilities and their ability to use it. Web-based applications offer a good solution to this problem because they allow the developer to utilize modern technology in model design while exercising full control of the user's interaction with the model.

This report seeks to illustrate this idea in the context of reservoir release scheduling. Reservoir operators are faced with the challenge of making decisions in real-time while balancing competing demands with uncertainty in the future. It is essential that reservoir operators have access to models that facilitate good decision making. Often, it is not necessary that the operator know the nuanced details of a model; rather, they are more concerned that the model is accurate, dependable, and usable. The objective of this report is to evaluate the feasibility of using a web-based application to schedule reservoir releases to meet a downstream target. To accomplish this object, a prototype web-application model was developed to assess feasibility, or the accuracy of the application, its optimization performance, and the ability of the web-based application to perform reservoir scheduling. This report will provide a discussion of the different routing, optimization and programming methods utilized to build the prototype model, and (2) providing analysis to illustrate the feasibility of utilizing the application for reservoir release scheduling.

2. Application Basin

2.1 Area of Interest

The months of April through July in the Yampa River basin of northwestern Colorado mark the period of the year when snow accumulations on the western slope of the Colorado Rockies melt and runoff into the river. As the Yampa River traverses some 160 miles from its headwaters in the Rocky Mountains through the high desert of northwestern Colorado, its flows are both supplemented by several snow melt driven tributaries and diminished by both natural losses in the river and diversions from the river to support agriculture in small municipalities along its channel. The Yampa River terminates into the Green River near the border of Utah and Colorado where it meets outflows from Flaming Gorge Reservoir roughly 60 miles upstream on the Green River (Google Earth, 2021).

Reservoir Operators of Flaming Gorge Reservoir are required to make releases from the reservoir to meet downstream flow targets at the USGS gage near Jensen, Utah, which is roughly 25 miles downstream of the confluence of the Green and Yampa Rivers. To operate the reservoir optimally, enough water needs to be released from Flaming Gorge reservoir so that the target is met; however, if too much water is released, operators would have depleted storage in the reservoir that was not required to be released. Operating the reservoir optimally is difficult for a combination of two reasons. The first is that travel times from Flaming Gorge reservoir to the Jensen Gage are long, as the reservoir is located roughly 85 miles upstream of the gage (Google Earth, 2021). The second is that, during runoff, the Yampa River exhibits diurnals in flow due to

snow melt, and timing Flaming Gorge Reservoir releases to combine with the diurnal flows on the Yampa is challenging (Nation Water Information System: Mapper, 2021).

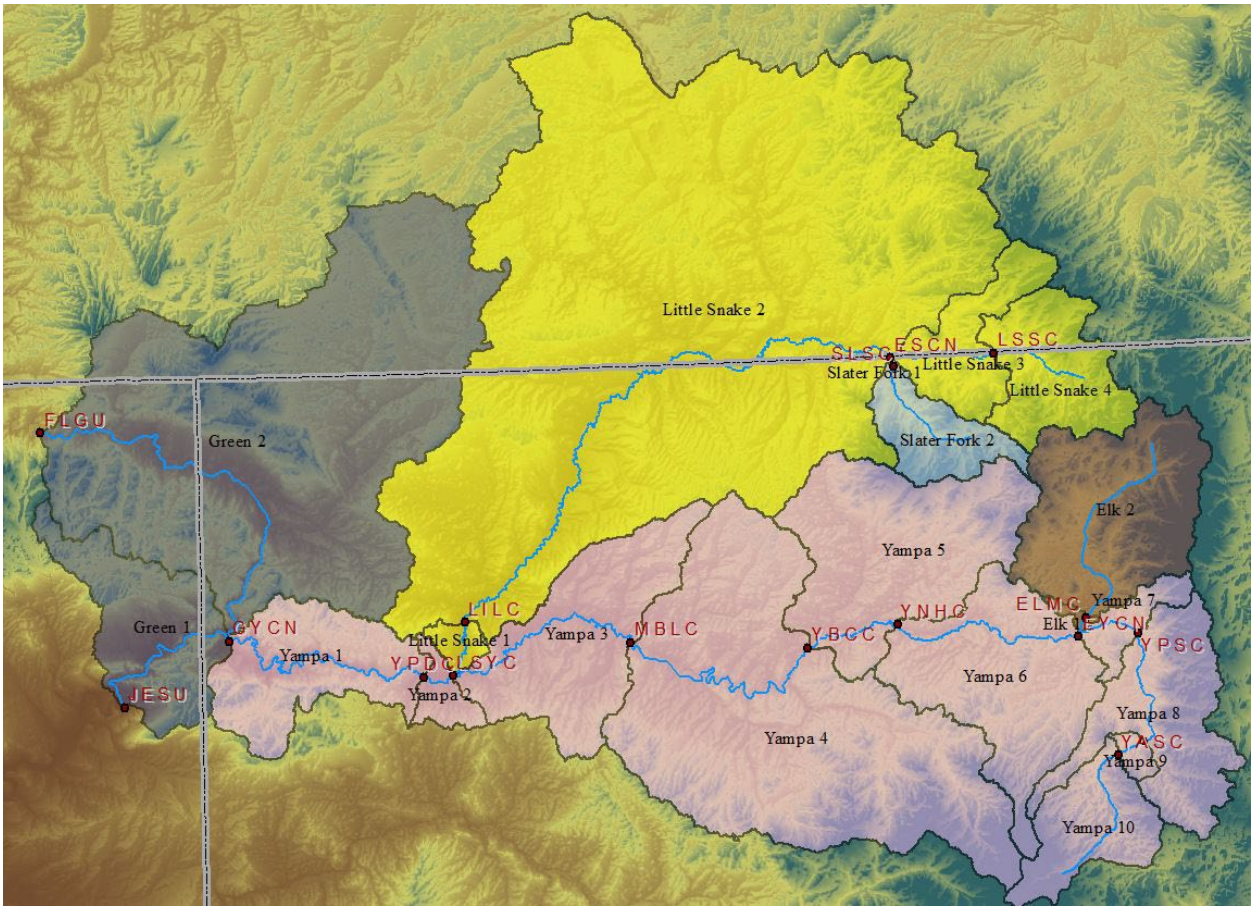

Figure 1: Subbasin Delineation for the prototype model and the Hydromet Identifier of USGS gages used within the model (Coors, User Manual, 2003). Pink represents subbasins strictly in the Yampa Basin, Yellow represents subbasins of the Little Snake River (tributary to the Yampa River). Brown represents subbasins of the Elk River (tributary of the Yampa River). Blue represents subbasins of the Slater Fork River (tributary of the Little Snake River). Grey represents subbasins of the Green River.

2.2 Origins of the Prototype Model

The web-based application prototype model has its origin from an Excel model in the Yampa and Green River Confluence region that was similarly designed to help river managers make good decisions. While this Excel tool was effective computationally, it was found that it never became fully adopted because of the challenges it faced in user friendliness that are inherent to

Microsoft Excel (Coors, Routing Method Discussion, 2021). The Prototype Model catalogued in this report extends the original Excel Model by leveraging the flexibility of building a tool in a web-based environment to allow for both complexity and usability in the design the tool. More specifically, the Prototype Model extends the Excel Model by improving (1) the Optimization Method, (2) the model's resilience to breaking, (3) the user-friendliness of the model, (4) the forecast capabilities of the model, and (5) opening the door for more complex future development.

2.3 Structure of the Prototype Model

The prototype model divides the region surrounding the confluence of the Green and Yampa River basins into 14 subbasins (Figure 1), and subbasins were determined in accordance with the routing method that was utilized for the model (see section 3.1). The prototype model accounts for two tributaries to the Yampa River (the Elk River, and the Little Snake River) and the Slater Fork River tributary of the Little Snake River. Table 1 provides a summary of the prototype model's subbasin structure for each river in the model as well as the input gage associated for each river reach.

Table 1: Summary of the Prototype Model structure for each river in the Model.

River Basin	Number of Subbasins	Tributary Of	Input Gage of Upstream Subbasin
Yampa River	7	Green River	YPSC
Elk River	1	Yampa River	ELMC
Little Snake River	3	Yampa River	LSSC
Slater Fork River	1	Little Snake River	SLSC
Green River	2	Colorado River	FLGU

On the Yampa River, there are seven subbasins that drain directly to the Yampa River in the prototype model. The inlet to the most upstream basin on the Yampa (Yampa 7) is in Steamboat Springs, CO, and the USGS Gage 09242500 (YPSC) associated with this location provides input data for this subbasin to the prototype model. There are two major tributaries to the Yampa modeled in the prototype. The first is the Elk River, and it has one subbasin associated with it in the prototype model (Elk 1). The USGS Gage 09239500 (ELMC) is located at inlet to Elk 1, and this gage provides input data to the prototype model. The second tributary to the Yampa River is the Little Snake River. It has three subbasins in the prototype model. The USGS Gage 09253000 (LSSC) is located at inlet of the most upstream subbasin on the Little Snake River (Little Snake 1), and it provides input data to the prototype model. The Little Snake River has one tributary, the Slater Fork River, and the Slater Fork River has one subbasin associated with it (Slater Fork 1). The USGS Gage 09255000 (SLSC) is located at the inlet of Slater Fork 1, and this gage provides input data to the prototype model. The prototype model divides the Green River into two subbasins. Flaming Gorge Reservoir releases enter directly into the inlet of the most upstream subbasin on the Green River (Green 2). The USGS Gage 09234500 (FLGU) provides input data to the prototype model for Flaming Gorge Reservoir releases. The outlet of the downstream subbasin on the Green River, Green 1, is the USGS Gage 09261000 (JESU), and it is the target location for reservoir releases.

2.4 Data Requirements

There are two required inputs to the prototype model, and one optional input. The first required input is four to ten days of hourly observed flows for each gage within the model. Gage data is utilized as input data to any subbasin where the inlet is a gage; that is, in the observed period, the routing method will route gaged inflows wherever possible in the system. Table 2 provides a summary of all the USGS gages utilized within the proto-type model. In total, eleven USGS gages are used (Nation Water Information System: Mapper, 2021). Five of these gages are used for input data to the inlets of the upstream ends of the river basins. Another five of these gages are used as input data to the inlets of interior subbasins of the model and to calibrate the model as water is routed downstream through the system. The most downstream gage, JESU, is used both for model calibration and as the control point target.

Table 2: Table of the Hydromet Identifier, USGS Gage ID, and USGS Gages Description of all river gages associated with the prototype model.

Hydromet Identifier	USGS Gage ID	Gage Description	Model Use
LSSC	09253000	Little Snake River Near Slater CO	Input Data of Upstream Subbasin
SLSC	09255000	Slater Fork Near Slater CO	Input Data of Upstream Subbasin
LILC	09260000	Little Snake River Near Lily CO	Input Data Interior Subbasin/Calibration Data
YPSC	09242500	Yampa River at Steamboat Springs CO	Input Data of Upstream Subbasin
ELMC	09239500	Elk River Near Milner CO	Input Data of Upstream Subbasin
YNHC	09244490	Yampa River Above Elkhead Creek Near Hayden CO	Input Data Interior Subbasin/Calibration Data
YBCC	09247600	Yampa River Below Craig CO	Input Data Interior Subbasin/Calibration Data
MBLC	09251000	Yampa River Near Maybell CO	Input Data Interior Subbasin/Calibration Data
YPDC	09260050	Yampa River at Deerlodge Park CO	Input Data Interior Subbasin/Calibration Data
FLGU	09234500	Green River Near Greendale UT	Input Data of Upstream Subbasin
JESU	09261000	Green River Near Jensen UT	Calibration Data / Control Point

The second required input is a schedule for one to five days of future releases from Flaming Gorge Reservoir. This is input time series is fine-tuned by user to determine reservoir releases that optimally meet the downstream target and Jensen, Utah.

The third and optional input to the model is one to five days of hourly forecasted flows that occur at the inlets for the most upstream subbasins in the model. This data is routed through the model and extends the outlook period of the model into the future. The prototype model incorporates four days of test forecast data. This time series was developed utilizing a simple, empirical

proprietary method developed in house at Precision Water Resources Engineering (PWRE) (Coors, Routing Method Discussion, 2021).

3. Methods

3.1 Routing Method

The routing method implemented in the prototype model is a proprietary and empirical hourly routing method developed by Shane Coors of Precision Water Resources Engineering (PWRE) in Loveland, CO (Coors, User Manual, 2003). The method was developed specifically for areas where flows are highly variable because, in these cases, conventional methods like Muskingum or Storage-Outflow methods showed inaccuracy and instability. The method was utilized by the Bureau of Reclamation (Upper Colorado Region) from 2010 to 2014 and has been used by the Federal Water Master of the Truckee River since 2010. In both cases, the routing method was used in a model to support high-precision real time hourly reservoir operations for achieving environmental flow targets and improved operational efficiency (Coors, Routing Method Discussion, 2021).

This method was implemented into the prototype model for two reasons. The first is that the method has proven to work effectively in real world applications. In the Truckee River System, this routing method has been utilized to develop travel times of reaches on the Truckee River at varying levels of flows. Developing these travel times has allowed river authorities to (1) more efficiently perform accounting on the river, and (2) make reservoir releases to meet downstream demands more precisely. Furthermore, this method is utilized within the Truckee River System to do real time flood operations during large precipitation events (Coors, Routing Method

Discussion, 2021). The second reason this method was selected is that there are only three parameters per reach of river in the method that need to be calibrated by the user, which makes operating the model and the optimization process within the model dynamic and responsive.

The following three sections provide a summary of how the routing method is implemented, its input and parameters, and some of its limitations.

3.1.1 Subbasin Delineation

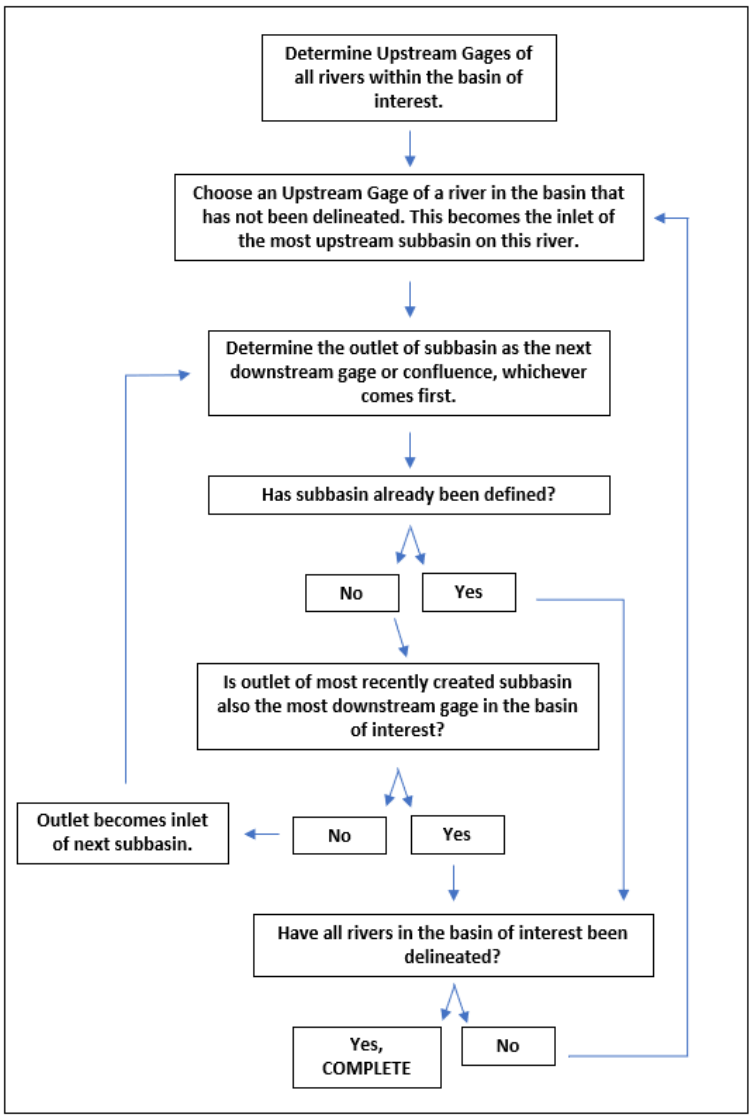

Figure 2: Decision tree for delineating subbasins in the routing model.

Figure 2 provides a decision tree diagram of the process of delineating subbasins using this routing method. It starts by locating the most upstream gages on all branches of the rivers in basin of application. These gages are the inlets to the most upstream subbasins on rivers in the model (i.e., inflows to the model). The outlet of the most upstream subbasin on a river is determined by the first occurrence downstream of either a gage or a river confluence. This outlet becomes the inlet for the next downstream reach, and the process of locating the next

downstream gage or river confluence is repeated. The process repeats itself until all river reaches in the basin of application are delineated into a subbasin.

Figure 3 illustrates two cases for delineating subbasins using this routing method. In the figure, the “inlets” can represent either a gage or confluence. In Case 1, the outlet of the subbasin is a gage and this requires only one subbasin to be delineated. In Case 2, the outlet of the subbasin is a confluence, and this requires that two subbasins will need to be delineated (Coors, User Manual, 2003).

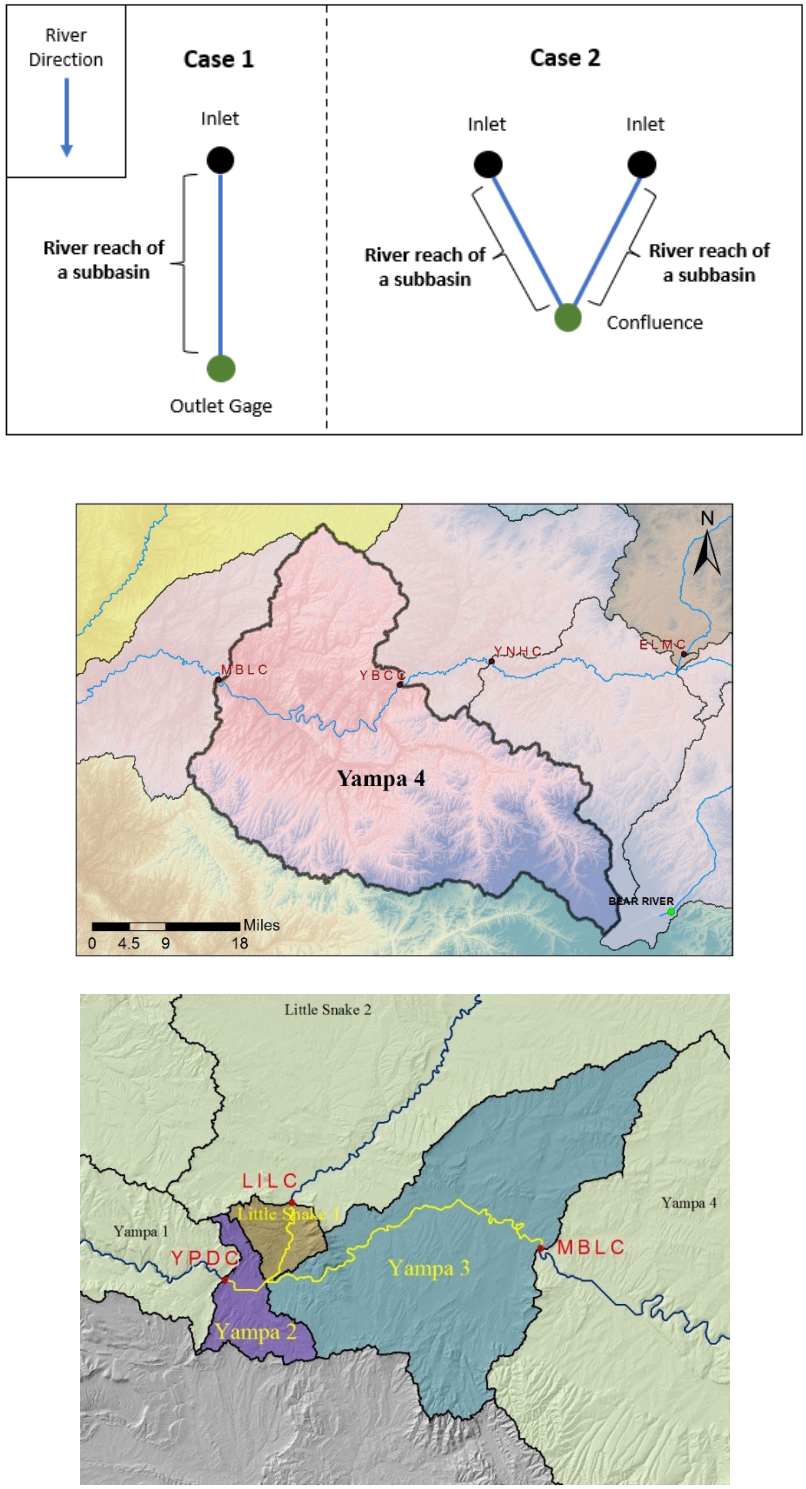

Figure 3: Top) Two cases for delineating subbasins in the routing method. Case 1 is the simple case where the outlet of a subbasin is a gage. Case 2 is the more complicated where there exists a confluence. Note that in Case 2, two subbasins are defined. Middle) Example of Case 1 in the Prototype Model. Bottom) Example of Case 2 in the prototype model.

The prototype model provides a good example of the subbasin delineation process. The ELMC gage on the Elk River is the upstream inflow gage to the model for this section of river. Downstream of this inflow gage, there is no other gaging location before it reaches its confluence with the Yampa River. Thus, this subbasin is defined by the land area that drains to the reach of the Elk River between the ELMC gage and the Elk River's confluence with the Yampa.

Figure 4: Illustration of the Elk 1 subbasin on the Elk River, one of the four most upstream subbasins in the application basin's river network. Note the gage, ELMC, at the inlet of the basin, and the confluence with the Yampa at its outlet (Coors, User Manual, 2003).

The next subbasin is on the Yampa River. Its inlet is the confluence of the Elk and Yampa River, and the outlet is the next downstream gage YNHC. This process continues for Yampa River downstream of its confluence with the Elk River until the JESU gage on the Green River is reached. At this point, another river is selected in the basin of application, and the process is resumed until all rivers have been delineated into subbasins. Figure 1 in Section 2 provides an image of the Yampa and Green basins fully subdivided.

3.1.2 Reach Routing and Method Parameters

The routing method accomplishes moving water through the system by routing water successively through each subbasin, starting with the most upstream subbasins and ending with the most downstream subbasin. When an upstream subbasin is routed, the outflow of this subbasin becomes the inflow to the next downstream subbasin. *If the inlet of the next downstream subbasin corresponds to a river gage, the downstream subbasin will route the observed data for this gage first, and then any water from the upstream subbasin that arrived after the most current gage reading.* Furthermore, if the upstream end of a subbasin is a confluence, this subbasin will not solve until both upstream subbasins have been routed.

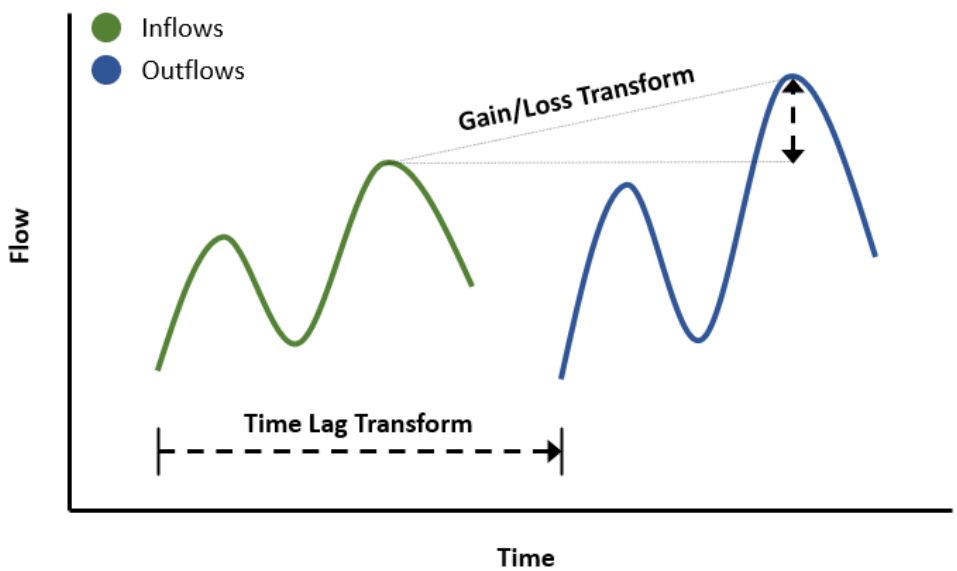

Figure 5: Illustration of the two main transforms performed on inflows by the routing method to produce outflows.

The routing method accomplishes the routing of water through a single reach of a subbasin by "Point Mapping". Each timestep and flow pair (t, O) is mapped by a two-step transformation to a new point $(t + \Delta t, O + \Delta O)$. This transformation process is summarized pictorially in Figure 5.

The first step is the "Time Lag Transformation", and its purpose is to lag the inlet flows of a subbasin to account for the travel time to the outlet. The second step is the "Gain/Loss Transformation", it scales the inlet flows of a subbasin to account for the gains and losses that occurred in the river reach. In special cases, a third transform, the "Smoothing Transform," is applied for reaches that exhibit large and abrupt changes in inflows (i.e., reservoir releases), or that have reaches that have very long lag times. These three transforms and the parameters associated with them are described in the proceeding sections (Coors, User Manual, 2003).

3.1.2.1 Time Lag Transform

The fundamental assumption of the Time Lag Transform is that as flow increases, the lag time, or travel time, of the flow through a reach of river decreases. This correlation between flow and lag time can be characterized by a power curve. The transform applies the following formula to determine the lag time in hours, for a reach's inflow to traverse through a river reach.

$$\boldsymbol{T_{Lag}} = A * \boldsymbol{Q_{In}^{B}} \tag{1}$$

In this equation, $\boldsymbol{Q_{In}}$ represents a vector of hourly inflows in cubic feet per second to a given reach, parameter A represents the lag coefficient, parameter B represents the lag exponent, and $\boldsymbol{T_{Lag}}$ represents a vector of output lag times in hours (i.e., the number of hours that an associated flow takes to traverse the reach) (Coors, User Manual, 2003).

The lag coefficient parameter, A, is physically based, and it is the reach length in meters divided by one thousand. Because the lag coefficient is physically based, it is not included in the set of calibration parameters in the routing method. The lag exponent, B, is empirical, and it is generally between -.45 and -.05 (Coors, User Manual, 2003).

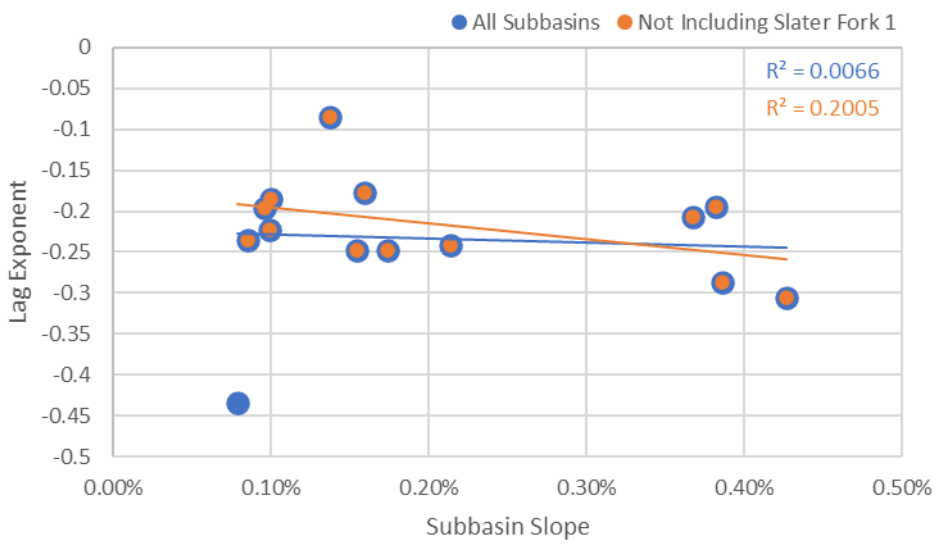

Figure 6: Comparison of subbasin slope percentage with the optimized Lag Exponent for the prototype model. The orange trendline corresponds to the trend excluding the outlier of the Slater Fork 1 subbasin. The blue trend line includes all subbasins.

Figure 6 shows a comparison of the average subbasin slopes and optimized lag exponents in the prototype model. The figure suggests that exclusive of one outlier, steeper subbasins tend to have a smaller lag exponent. The outlier is associated with the Slater Fork 1 subbasin, which is unique in that it has both a shallow slope of .07% and it only represents a 1.3-mile section of river, which is the shortest reach of river in the model. In comparison, the subbasin Elk 1 is a 3 mile stretch of river in a mountainous region with a steeper slope of 0.39%. The Elk 1 subbasin's optimized Lag Exponent is -0.28, which fits the trend of smaller Lag Exponents with larger

slopes. This suggests that the Slater Fork 1 subbasin might be an outlier due to the combination of its short reach length *and* shallow slope. Slater Fork 1 is also a part of the region in the model that calibrates the poorest, and this suggests that there could be limitations on the routing method's time lag capabilities in the cases of short reaches with shallow slopes. As the routing method was developed for use in snow-melt driven basins mountainous regions, future studies could improve the routing method by investigating the limitations of not only the subbasin slope on the routing methods performance but also river geomorphology and cross-sectional shape.

3.1.2.2 Gain/Loss Transform

The fundamental assumption of the Gain/Loss Transform is that because the method is intended to be utilized in snow-melt driven basins that exhibit diurnals in flow, lateral inflows to the subbasin will also exhibit diurnal flows and be correlated to the inflow to the basin. This correlation is described by the routing method in the following equation:

$$\boldsymbol{Q_{Out}} = C * \boldsymbol{Q_{In}^{D}} \tag{2}$$

In this equation, $\boldsymbol{Q_{In}}$ represents a vector of hourly inflows in cubic feet per second to a given reach, parameter C represents the scale coefficient, parameter D represents the scale exponent, and $\boldsymbol{Q_{Out}}$ represents a vector of output flows for the subbasin that will occur at the outlet at the time specified by the Time Lag Transformation. Both the scale coefficient and scale exponent parameters are empirical and not physically derived; however, preliminary analysis on the Gain/Loss transform has indicated that these two parameters are correlated to the area of the

subbasin and the average melt-out date of snow in the subbasin (Coors, User Manual, 2003). Typical values for the scale coefficient range from roughly 0.5 to 4, and typical values of the scale exponent range from -0.25 to 0.25. These values do change throughout the season and must be calibrated when the model is used. Furthermore, it is important, that in applying this method, proper boundaries are placed on these coefficients for each subbasin to prevent the model from "over-fitting" during calibration and optimization. These boundaries will be discussed more in Section 3.1.3 on Challenges of the Routing Method below.

3.1.2.3 Smoothing Transform

In some cases of subbasins that constitute both long reaches of river (i.e., greater than 50 to 60 miles) and/or are subject to discontinuous jumps in inflow (like reservoir releases), the routing method does not perform well without the addition of the Smoothing Transform. As illustrated in Figure 7, as pulses of river flows travel a reach of river, they tend to attenuate and disperse. The routing method handles this by applying a centered moving average of a selected number of hours to the outflows of a reach after the Lag Time and Gain/Loss transforms have been applied to the inflows. The selected number of hours is a method parameter called the smoothing hours (Coors, User Manual, 2003).

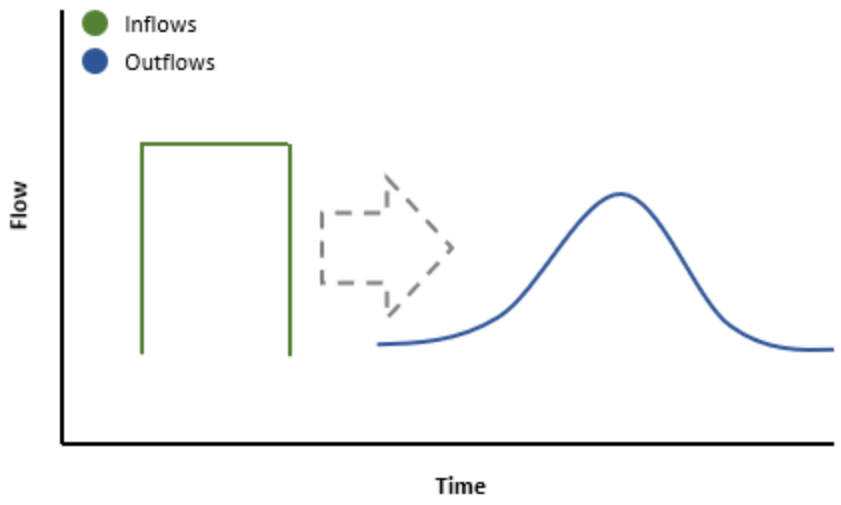

Figure 7: Illustration of the attenuation and dispersion effects on a pulse of water travelling through a reach of river.

As an example, in the prototype model it was found that only the Green 2 subbasin required a Smoothing Transform. After the model was built and calibrated with no Smoothing Transforms applied, the Green 2 reach performed poorly because (1) releases from Flaming Gorge Reservoir into the subbasins inlet were highly variable and discontinuous, and (2) the reach of river in Green 2 is over 60 miles long. Figure 8 shows the inflows to and the modeled outflows from the Green 2 reach for two cases: one where no Smoothing Transform applied, and a second where a Smoothing Transform using a smoothing hour parameter of 7 hours is applied. In Case 1, the model performed poorly and the Nash-Sutcliffe Model Efficiency Coefficient (NSE) for the next downstream gage was -0.71. In Case 2, the model performed very well, and the NSE for the next downstream gage was 0.8.

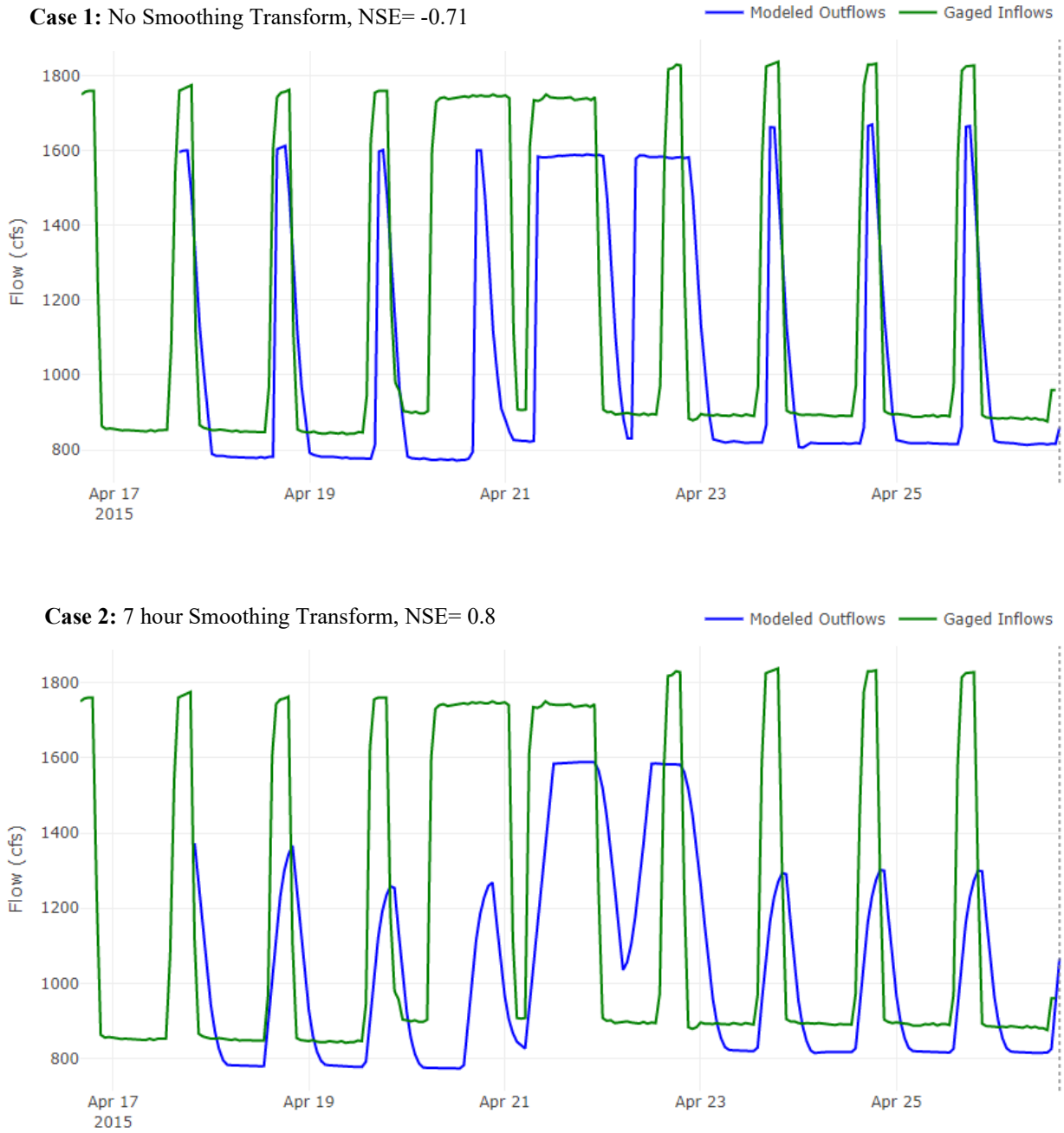

Figure 8: Modeled outflows of Green 2 subbasin after calibration with: Case 1) no Smoothing Transform applied, and Case 2) a Smoothing Transform applied with the smoothing hours parameter set to 7 hours.

3.1.3 Limitations of the Routing Method

3.1.3.1 Gage Proximity Limitations

The routing method utilized has several limitations that are essential to keep in mind when it is utilized. The first is that adequate gaging is necessary for the routing method to perform well. Long river reaches pose a problem for the routing method's "Point Mapping" transformations particularly when flows change rapidly. In these cases, differences in time lags between subsequent hourly data points can be more than an hour, resulting in an outflow hydrograph that is no longer in temporal sequence. This results in a later time and flow point "catching up" to an earlier flow point. Furthermore, long reaches of river introduce added complexity of larger losses or gains due to groundwater exchanges and more inconsistency in diversions from the river that occur on the river reach. This poses challenges for the Gain/Loss Transform of the routing method.

Figure 9: Subbasin map of Little Snake 2 that spans roughly 112 miles of the Little Snake River. Its upstream end is the confluence of the Little Snake and Slater Fork Rivers. Note, the Slater Fork 1 subbasin is small (roughly 1 mile), and the SLSC gage is at the inlet of the Slater Fork 1 subbasin.

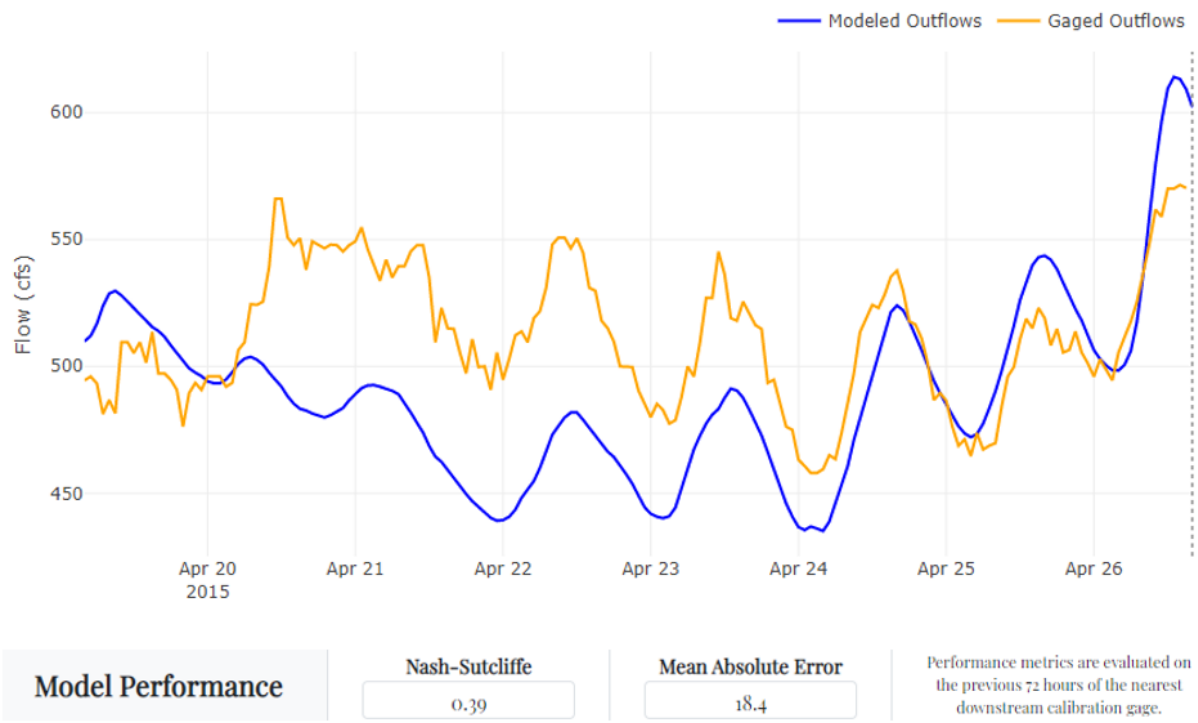

Figure 10: Modeled outflows from the Little Snake 2 subbasin verses actual outflows. After optimizing the model to the last 72 hours of observed data, this subbasin performed the worst in the model with an NSE of 0.39.

To provide an example, the prototype model's subbasin of Little Snake 2 constitutes a 112-mile reach of the Little Snake River (Figure 9). It has a gage at its downstream end (LILC), but its upstream end is the confluence of the Little Snake and Slater Fork Rivers. The upstream subbasins to Little Snake 2 are Little Snake 3 (15-mile river reach) and Slater Fork 1 (1.3-mile river reach). As shown in Figure 10, the model struggles to calibrate these three subbasins to their shared downstream gage at LILC. The model calculates an NSE of 0.39 for 72-hour period leading up to the last observed date, and the model performance diminishes greatly outside of the optimization period, specifically between midday on April 20th and end of day on April 23rd. The poor performance of this section of river is likely due to its lack of gaging for longer stretches of

river and the possibility that diversions were inconsistent and changing within the optimization period.

In summation, this routing method does struggle with subbasins that contain long river reaches. The prototype model of the Yampa and Green rivers suggests that the routing methods performance diminishes with river reaches longer than 65-80 miles, and adequate gaging needs to be a selection criterion for this routing method. Furthermore, river reaches are more accurately modeled in regions of river where diversions are not as common. The routing method could be enhanced to incorporate diversions into it, but further studies would need to be conducted on how to incorporate this effectively. Furthermore, this would add to the complexity of the model as it would require input diversion data.

3.1.3.2 Limitations as an Empirical Model

The second limitation of the routing method is that it is an empirical method. Of its four main parameters to route water through a reach of river, only the Lag Coefficient is known to be physically based. A drawback to using an empirical method is that, without careful attention, it allows for overfitting the model to observed conditions. Constraints are put on parameter ranges in the method (these constraints are discussed more in Section 3.2.2 on Optimization Constraints); however this does not always prevent overfitting from occurring, and it is essential that users of the method are very familiar with the basin of interest so that they can identify when overfitting may be occurring. In the case for the objective of this project, it was determined that users of the model (reservoir operators) would have a good understanding of the basin that is modeled. This allows for quality control on the tool to be sure that optimization solutions and

calibration provide results that are physically realistic when using the model, even if this comes at the expense of model performance.

While there are limitations to the method being an empirical method, they do not negate its ability to provide useful predictions in the context of this model. It is important to note that the model is always calibrated to the previous 72 hours of flow, which, when calibrated well, are good indicators of what will happen in the proceeding several days *at the gages*. The key is that the method will predict well *what the gage* will measure since it is the gages that the model is calibrated to. It will not necessarily predict the actual physical flows because gages are inherently inaccurate. Given that Reservoir Operators are concerned with what the *gages* are measuring and that this method can (1) model recent behavior of the river inclusive of gaging error optimally, and (2) project it forward provides evidence that, despite its limitations, it is a good method to use within the context of this problem.

3.2 Optimization Algorithm

The optimization method utilized in the prototype model is the "Sequential Least Squares Programming (SLSQP) Algorithm" provided by the open source SciPy Python library (SciPy.org, 2021). This method is a subsect of Sequential Quadratic Programming that allows for linear constraints (Yang, 2017). This method is appropriate for three reasons. The first is that it solves non-linear, smooth objective functions. The object function utilized in the prototype model is the Nash-Sutcliffe Model Efficiency Coefficient (NSE) and it is a non-linear smooth function. The second reason this method is appropriate is that it allows for linear constraints. The constraints of the objective function are bounds on the Time Lag and Gain/Loss transform

parameters for a given reach, which are linear. The third reason this method is appropriate is that it performs well in optimizing the system.

The proceeding three sections discuss in detail the organization of optimization within the model, the optimization constraints, and the objective function.

3.2.1 Optimization Method Organization

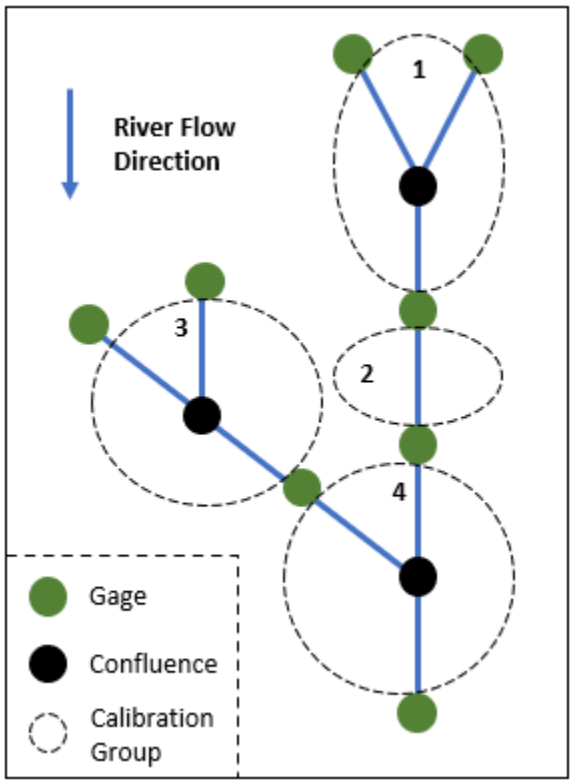

Figure 11: Depiction of how subbasins of the basin of interest are organized into calibration groups.

To apply optimization methods to the model, the basin must be divided into calibration groups, or groups of subbasins that share a downstream gage, necessitating that these subbasins be calibrated simultaneously. Figure 11 provides an illustration of how the basin is subdivided into

calibration groups to implement the optimization (Coors, Routing Method Discussion, 2021). As shown in the figure, calibration groups contain either one or three subbasins in them. In the figure, Calibration Group 2 is a calibration group with one subbasin in it because both the inlet and outlet of the subbasin corresponds to a gage. In this case the optimization method will optimize the three empirical parameters associated with the subbasin in the calibration group: the lag exponent, the scale coefficient, and the scale exponent. The lag coefficient is excluded from the optimization because this parameter is physically based on the length of the reach in the subbasin. It was determined that the optimization method performed better when all three parameters were optimized simultaneously, as opposed to first running an optimization on the Time Lag Transform component of the routing method, and then the Gain/Loss Transform of the routing method. Calibration groups 1, 3, and 4 in Figure 11 are comprised of subbasins that are connected by a confluence that share a downstream gage. Each calibration group has 3 subbasins associated with it. For each of these calibration groups, the optimization method will calibrate the three subbasins contained the calibration group simultaneously. Therefore, the optimization method will be optimizing a total of nine parameters: the lag exponent, the scale coefficient, and the scale exponent for each of the three subbasins. It was determined that the optimization performed more effectively when all nine parameters were optimized simultaneously, as opposed to first optimizing the upstream subbasins, and then optimizing the downstream subbasins.

3.2.2 Optimization Constraints

One of the required inputs of the SLSQP optimization method are constraints on the input variables. These constraints correspond to upper and lower bounds for the lag exponent, the scale coefficient, and scale exponent parameters of the routing method for each subbasin included in

the calibration group to be optimized (Coors, User Manual, 2003). As described in Section 3.1.3.2, because the model is an empirical model, careful attention must be shown to the model results to avoid overfitting the model and providing results that are not physically realistic. These constraints are important, as they provide one way to prevent the optimization from over fitting the observed data and giving physically unrealistic results. An example of this can occur in the case of a calibration group that contains a confluence. In this type of calibration group, the modeled outflows of two subbasins provide the inflows to the downstream subbasin. Without constraints on the input parameters, the optimization would be permitted to transform the inflows of the two upstream subbasin however it needed so that their combined outflows could be routed through the downstream subbasin optimally. The problem with this is that what may fit the outlet gage best is to solve that one upstream subbasin is gaining an unrealistic amount of water as it is routed through its subbasin and the second upstream subbasin is losing an unrealistic amount of water when routed through its subbasin. Many of these issues, but not all, are circumvented by applying constraints on the parameters' ranges. A detailed example of this occurring is provided in Section 4.2 on Optimization Performance.

The process of picking the constraints on the lag exponent, scale coefficient and scale exponent parameters for calibration groups with a multiple subbasins in them is challenging. The inherent issue is that the two upstream basins in the calibration group do not have a downstream gage. The development of these constraints requires a combination of two things. The first is a good conceptual understanding of what is occurring on the ungagged subbasins (i.e., should flows be gaining, losing, or remaining unchanged) (Coors, Routing Method Discussion, 2021). This may include what diversions are occurring or what the lateral inflows and losses look like at different points in the year. The second is time for the model to mature over the course of several

years through different conditions (Coors, Routing Method Discussion, 2021). The case of picking constraints on the parameters for calibration groups with a single subbasin in them is more trivial, as both the inflows and outflows are gaged. Again, model maturity helps to define these ranges as the model sees more data. Obtaining an accurate value for these constraints is the most challenging piece to making the optimization process more complete, and the importance of checking optimization results with knowledge about how the basin is known to physically behave is emphasized.

3.2.3 Optimization Objective Function

The objective function that is utilized calculates the Nash-Sutcliffe Model Efficiency Coefficient (NSE) based on the difference between modeled flows and observed flows at the downstream end of the calibration group. The NSE was utilized as the result of the objective function because it amplifies the existence of large errors between modeled and observed flows, and in timing reservoir operations the existence of large errors have more significance (Niemann, 2020).

The objective function takes as input the optimizable parameters that the optimization algorithm provides it for the subbasins in the calibration group being optimized. It also takes the static input of the hourly time series of observed inflows to the upstream ends of the calibration group, the observed outflows of the downstream end of the calibration group, and the lag coefficient associated with each reach. Lastly, it takes as input the number hours to evaluate model performance preceding but inclusive of the last observed flow in the flow. Currently, this input is set to 72 hours with the assumption that calibrating to the past 72 hours will allow the optimization method to calibrate the model to the most recent river conditions. Each time the

objective function is called, it takes the input parameters given to it by the optimization algorithm and utilizes them to route the observed inflows to the calibration group through calibration groups subbasin network to calculate modeled outflows. The objective function then takes these modeled outflows and calculates the NSE using the following equation

$$NSE = 1 - \frac{\sum_{t=1}^{T}(Q_m^t - Q_o^t)^2}{\sum_{t=1}^{T}(Q_o^t - \bar{Q}_o)^2} \tag{3}$$

where

NSE, *Nash-Sutcliffe Efficiency*

t = 1*, the first timestep of the optimization period*

T, *the timestep of the last observed flow*

Q_m^t, *the modeled outflows at time t*

Q_o^t, *the observed outflows at time t*

$\bar{Q}_o$, *the average of the observed outflows over the optimization period*

Once this is calculated for the current set of inputs, the optimization method will determine how to further maximize the NSE (i.e., improve the model score closer to the maximum NSE of 1) by altering the input parameters.

3.3 Programming Tools Used for Implementation

Development of the web-application was driven by the following goals:

1. To implement the model using a modern technological infrastructure, allowing the ability to utilize access to modern computing techniques through open-source code.
2. To develop the code for the model generically for the prototype models basin such that reimplementation of the prototype could be easily facilitated.
3. To develop organized and easily readable code so that future adaptations to the model could be incorporated with minimal effort.

These goals facilitated the decision process for how the model would be implemented as a web-based application; the frameworks and coding languages used to implement the model provided a balance of flexibility and customization that allowed these goals to be achieved.

Implementing the prototype model as a web-based application required three main development phases. The first was to implement the routing method, optimization algorithm, and the reservoir releases scheduling into a Python based infrastructure (python, 2020). The second phase was to develop a back-end structure within the Django Python web framework (django, 2021), and to build Application Programming Interfaces (APIs) utilizing the Django REST Framework (django REST framework, 2021). The third phase was to develop a frontend web browser interface using the Vue JS single page application framework (Vue.js, 2021). Figure 12 illustrates the high-level structure of and relationships between the three phases of the prototype web-based application.

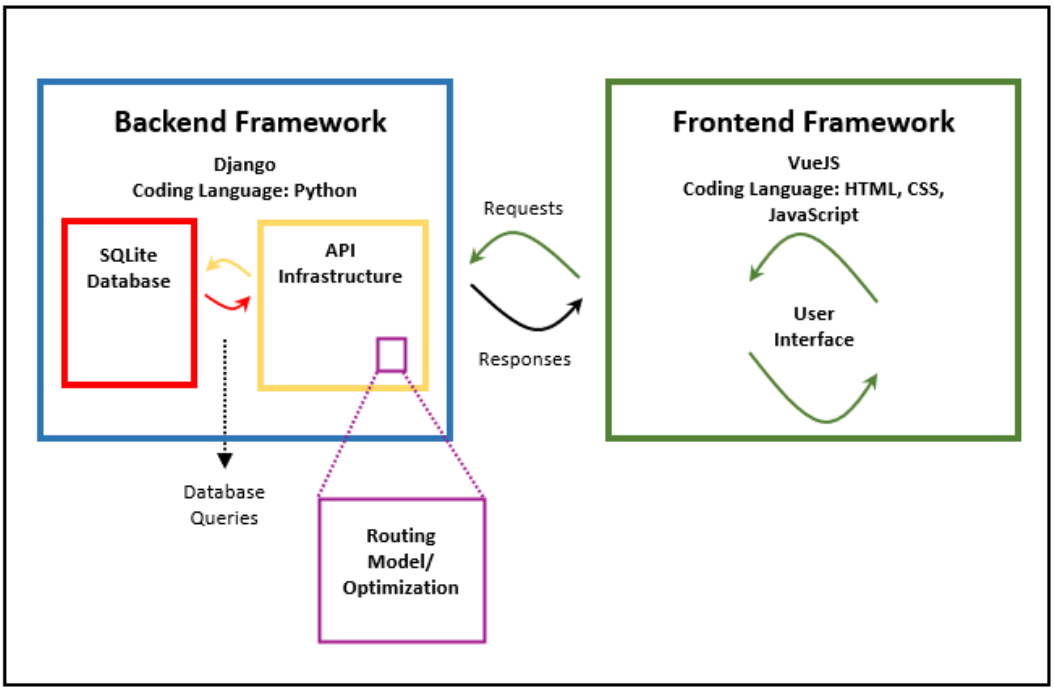

Figure 12: High-level structure and relationships between the Routing Model/Optimization Python Code, the Backend Framework, and the Frontend Framework

3.3.1 Routing Method and Optimization Algorithm

The routing and optimization mechanisms were implemented into the prototype model by utilizing the Python coding language. The Python language allowed for an adaptable code infrastructure by utilizing its abilities to develop Object-Oriented code. For example, when the code is run, Reach objects are created. Upon creation, Reach objects are assigned attributes, such as the reaches name, its upstream reaches, its downstream reaches, and its related gages (if applicable). Furthermore, once the Reach object is created, you can apply methods (or actions) to it). For example, you can call a method to route the reach, which will apply the routing method to the reaches inflow data and return the reaches outflows and their associated datetimes. A major strength of this structure is that it allowed for the ability to reduce redundancy in the code, which reduces complexity in the code and facilitates making changes to the code in the future.

A second strength of utilizing Python is that it allowed for the use of open-source libraries to accomplish algorithms necessary for reach routing and optimization. For example, the Numpy library allowed the reach routing to be accomplished in a few lines of code through its ability to apply vector math operation (Numpy v1.21 Manual, 2021). A more complex operation necessitated by the model was the implementation of the SLSQP algorithm to optimize the system. This algorithm was made easily accessible by implementing the Scipy library into the code, and it allowed the optimization component of the model to be coded in a clear and organized manner (SciPy.org, 2021).

Lastly, the Python language facilitated the ability development of code particular to hydrologic systems. An example of this was developing code to determine when a subbasin should solve for outflows based on of its inflows had been solved for (i.e., if all its upstream subbasins has been solved already). A second example of this was the ability to develop a recursive function to determine the calibration group that a particular subbasin was in based on the physical connectivity and gaging network of the basin system.

3.3.2 Backend Framework

The backend structure and functionality of the prototype model was built utilizing the Python based Django Framework and utilizing appropriate levels of code abstraction provided by the Django REST to facilitate the processes needed to run the web-application (django, 2021) (django REST framework, 2021). The Django and Django REST Frameworks facilitated the creation of a backend server to communicate information from the SQLite database and to compute model results via Application Programming Interfaces (APIs) to the frontend

framework. The main purpose of the backend framework of the prototype model is to listen and respond to instructions that the frontend makes of it. One of many examples of this would be the frontend requests that the backend runs the model with updated parameters for a reach and return the model results.

The first step in developing the backend in Django was to organize a database structure. It was quickly discovered that the connectivity properties of a river basin were well suited for a relational database. What is meant by the term “connectivity” is, in reference to the prototype model, the way a subbasin is connected to its surrounding subbasins. For example, given a subbasin, “connectivity” answers the following questions:

- What is the upstream subbasin? Are there multiple upstream subbasins?
- What is the downstream subbasin?
- Does the subbasin have an inflow/outflow gage?

Organizing the connectivity of code could pose a challenge; however, the principles of a relational database made these challenges easy to manage.

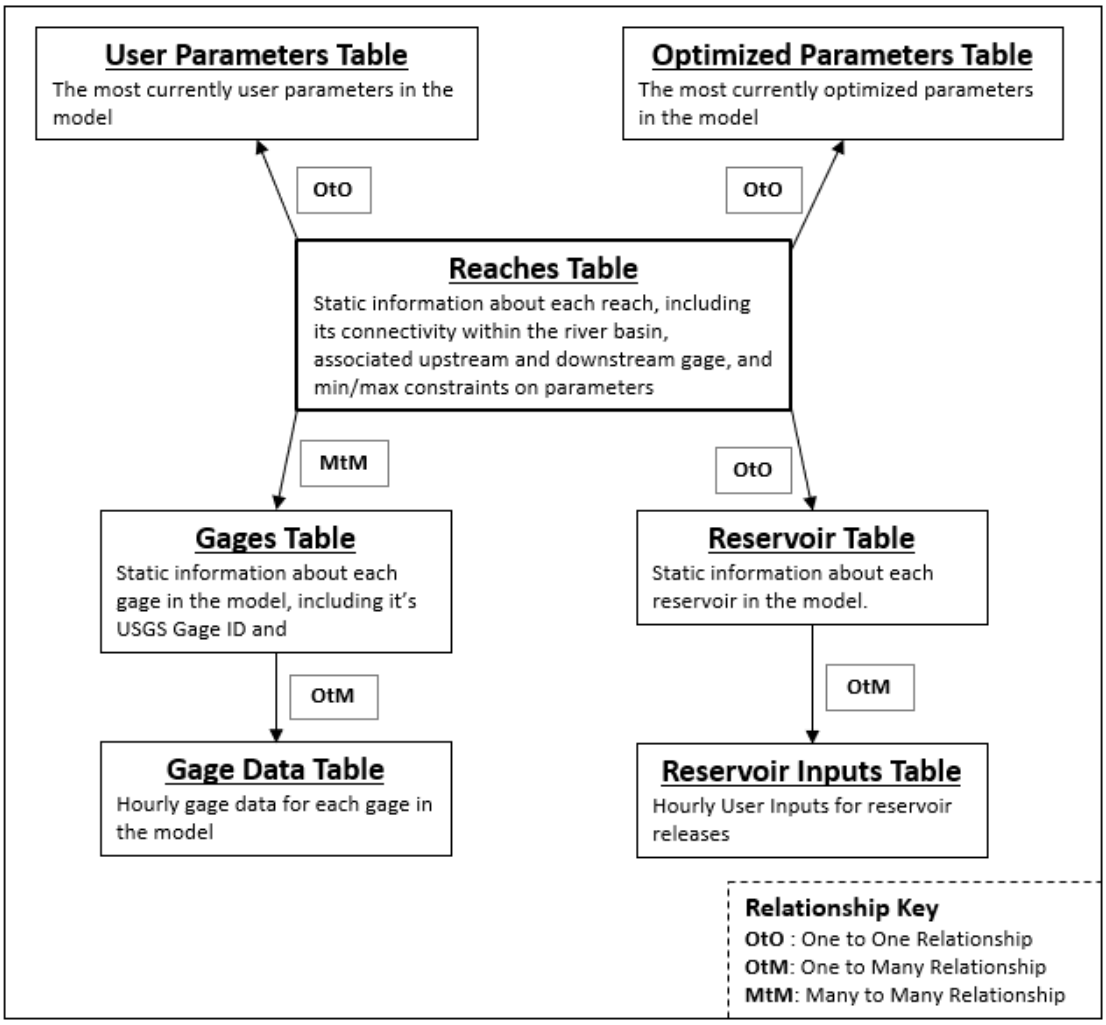

Figure 13: Organization of the SQLite database tables and the relationships between the tables within the Django Framework.

Figure 13 provides a diagram of the structure of the database tables and the relationships between the tables of the prototype model. The central aspect of the database structure is the "Reaches Table" which contains all the information about the connectivity of the system, the location of gages in the system relative to the reach/subbasin, and the static parameters associated with the subbasin. All other tables within the database, like gages and gage data, or reservoir inputs, have a defined relationship to the reaches table. This structure facilitates making data queries within the backend framework with ease.

The second step in implementing the backend was designing the infrastructure to handle requests for information from the frontend and returning a response from the backend to the frontend. Qualitative descriptions of the types of these requests from the frontend include but are not limited to:

- Provide the subbasin information for the currently selected reach.
- Update the parameters for a specific reach, run the model, and return the model results.
- Optimize the calibration group associated with the currently selected reach, update the optimization parameters stored in the database, and return the optimized results

To handle these requests and provide the appropriate response (i.e., build an API), the Django REST Framework was utilized and facilitated to accomplish to complex processes behind communicating information between the backend and frontend in a few lines of code.

An important aspect to the backend design is that model results are not stored in the database. It was discovered that the model run time to solve the entire system of the prototype model is only about 0.1 seconds. Because of the short run time, it was determined that running the model anytime a user made a request would not diminish the user experience of responsiveness of the web-application. The Django REST Framework package allows for the Python based routing model to be utilized within the backend structure to provide the necessary responses with ease.

3.3.3 Frontend Framework

The frontend framework is built using the Vue.js open-source JavaScript framework (Vue.js, 2021). Vue.js provides developers with a means to developing reactive single page applications, which made it an appropriate framework to utilize in building the prototype model. The

framework allows the data behind webpages to update without having to refresh the page, which enhances the user experience. One of the main benefits to using the framework is its organization. The inherent structure of the framework keeps the varying pieces of HTML, CSS, and JavaScript needed to build a frontend application well organized. Furthermore, the framework allows the developers to utilize what is called components, or reusable parts, of a webpage. Components are analogous to functions in other programming structures, and allow the code to avoid becoming repetitive, and thus making it easier to change and develop further.

Vue.js was utilized in the prototype model to build a web interface for the model user to interact with. Unlike programs like Microsoft Excel, developing the interface on a reactive web-based platform gives the developer full control over what the user can/cannot see or do. This allows the developer to build in "guard-rails" for the application, so that it is not misused or broken once in production. The user interface of the prototype model is divided into two main pages meant to be utilized in succession when operating the model. The first is the Reach Routing page. From this page, the user can run the model, view model results, calibrate the parameters, and view performance metrics to analyze how well the model is performing. Once calibration is finished, the Reservoir Ops allows the user to input reservoir releases, set reservoir targets, and run the model to see if input schedules will be routed through the system to meet downstream demands.

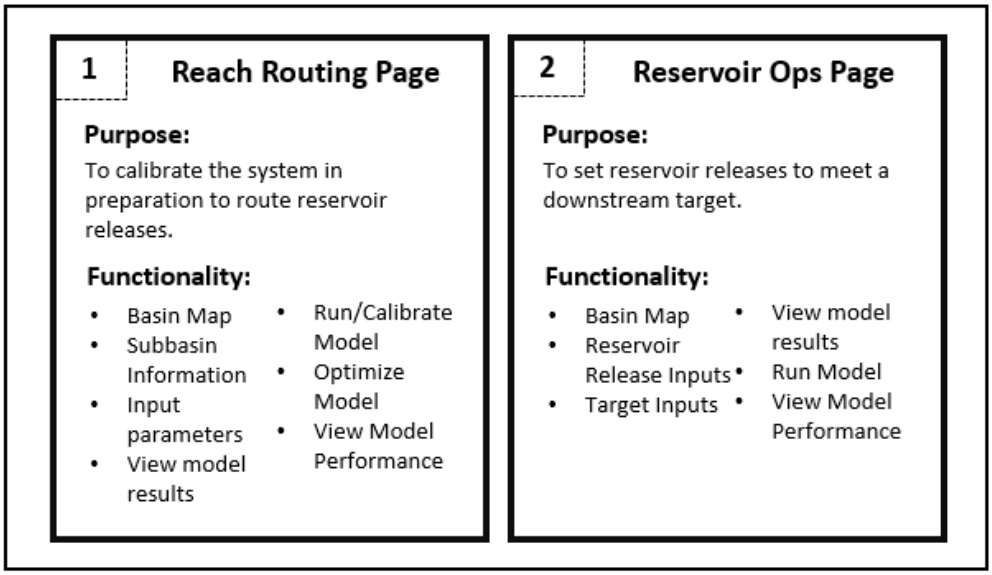

Figure 14: Summary of the purpose and functionality of the web-based user interface for the prototype model.

4. Model Analysis

To assess the effectiveness of the reservoir routing tool, the prototype model was loaded with hourly gage data for all gages within the model from April 17th, 2015, to April 26th, 2015. The latest observed data on April 26th was at 3pm, and this datetime is referred to as the "current time" of the model. The following three sections utilize this data to provide analysis on the routing method, the optimization performance, and the use of the tool for reservoir routing.

4.1 Routing Method Analysis

To assess the performance of the routing method utilized in the prototype model a comparison was made between it and the more established Muskingum Routing Method. This analysis was performed on the reach of river associated with the Yampa 4 subbasin which has a USGS gage at both its inlet and outlet (see Figure 1).

The prototype model was utilized to perform optimization on the parameters associated with the Yampa 4 subbasin. The optimization results for these parameters are shown in Table 3, and the time series modeled outflows utilizing the optimization parameters are plotted with the observed inflows and outflows to the subbasin in Figure 15. As shown in Table 4, the Prototype Model Method utilizing optimized parameters produced an NSE of 0.89, which is very strong. Furthermore, the MAE was only 18.17 cfs, which is relatively small in comparison with the magnitude of flows that were observed (between 1,800 and 2,200 cfs). As shown in Figure 15, during the 72-hour calibration period the Prototype Model Method solution for outflows performed decently at modeling diurnal peaks; however, it was not very accurate at modeling the duration of the peak. Furthermore, the timing of peaks flows of diurnals utilizing the Prototype Model Method tended to be slightly earlier than what was observed. For the troughs of diurnal flows, the Prototype Model Method tended to underestimate the actual flow, and the time of trough flows tended to be slightly later than what was observed.

Table 3: Optimized parameters in the prototype model for the Yampa 4 subbasin.

Prototype Model Parameters	**Optimized Yampa 4**
Lag Exponent	-0.22
Scale Coefficient	3
Scale Exponent	-0.12

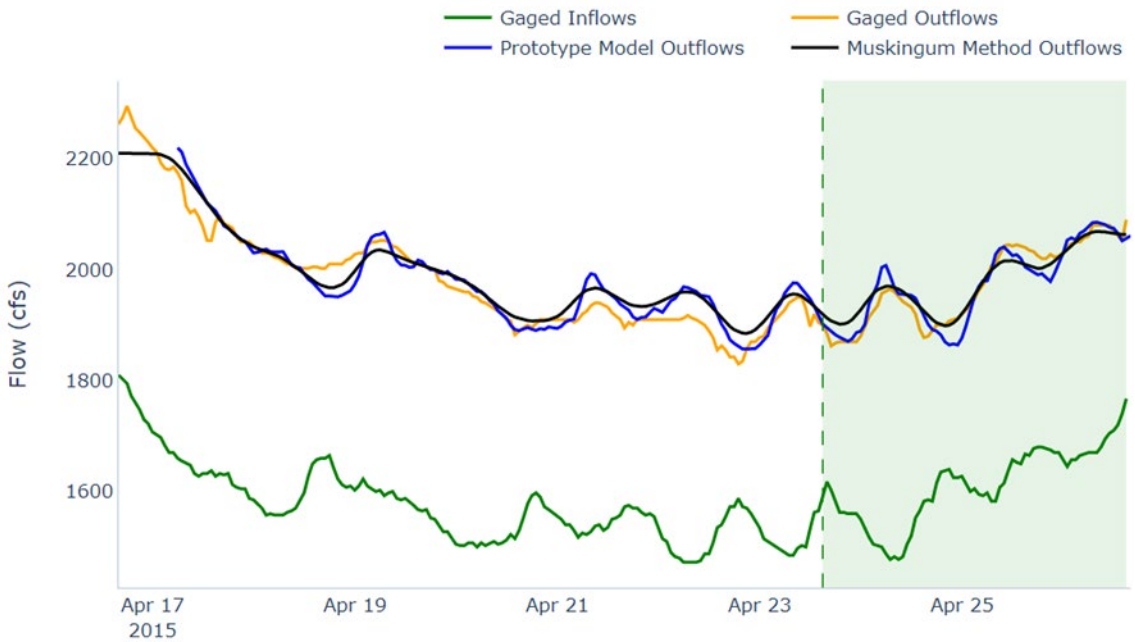

Figure 15: Plot showing the hourly time series of the gaged inflows and outflows, the optimized Prototype Model Method Outflows, and the optimized Muskingum Method Outflows for the Yampa 4 subbasin during the calibration period (green shaded region).

Table 4: Model Performance Metrics for the 72-Hour Calibration Period for Yampa 4 utilizing optimized parameters for the Prototype Model Method and the Muskingum Routing Method.

Performance Metric	Prototype Model Value	Muskingum Method Value	Difference
NSE	0.89	0.91	0.02
MAE	18.17	17.83	-0.34

The Muskingum Method was implemented for the Yampa 4 subbasin utilizing the HEC-HMS software developed by the United States Army Corps of Engineers (USACE). The Muskingum Method requires three parameter inputs. The first is the K parameter, or the travel time through the reach. The second is the X parameter which is a dimensionless quantity between 0 and .5 that lacks a strong physical meaning. The third parameter is the Number of Subreaches, which affects the routing attenuation (Applying the Muskingum Routing Method, 2021). The Muskingum Method does not have a direct way to apply gains that occur within the reach that is modeled,

and, as shown in Figure 15, the Yampa 4 subbasin gains a considerable amount of flow as water traverses its reach. To model this, an additional method was introduced within HEC-HMS to allow a constant gain to occur through the reach.

Table 5: Optimized parameters for both the Muskingum Method and Constant Gain for the Yampa 4 Subbasin.

Muskingum Optimized Parameters for Yampa 4	
K (Hr)	13.36
X	0.16
Number of Subreaches	9

Constant Gain Optimized Parameters for Yampa 4	
Discharge (cfs)	400

The optimized parameters to implement the Muskingum Method with a constant gain are shown in Table 5, and the time series modeled outflows utilizing these optimization parameters are plotted with the observed inflows and outflows to the subbasin in Figure 15. As shown in Table 4, the utilizing the Muskingum Method with optimized parameters resulted in an NSE of 0.91, which is slightly better than the prototype model performance. Furthermore, the MAE was only 17.83 cfs, which was 0.34 less than that the of the prototype model. Overall, the Muskingum Method tended to model peak flows of diurnals slightly more accurately than the Prototype Model Method; however, it different from the Prototype Model Method results in that it tended to overestimate the trough flows. The Muskingum Method shows similar behavior as the Prototype Model in modeling the timing of the peak and trough flows of diurnals: peak flows tended to occur slightly earlier that what is observed, and trough flows tend to arrive later than what is observed.

One characteristic of the Muskingum Method that is different from the prototype model is the smoothness of the curves. The Muskingum Method produces an outflow hydrograph that is extremely smooth, while the outflow gage shows more abrupt fluctuations in flow. While the prototype model method does not perfectly capture these fluctuations, it does provide an outflow hydrograph that is not smooth and shows more similar characteristics in shape to the outflow gage.

In summation, both methods performed well at modeling the outflows of the Yampa 4 subbasin, and Muskingum Method only slightly outperformed the Prototype Model Method. This suggests that the Prototype Model Method, while not well established, is a viable method to utilize when performing reach routing.

4.2 Optimization Analysis

4.2.1 Optimization Functionality

The optimization method generally performed well in identifying the parameters for each reach that maximized the NSE for the most recent 72 hours of gage data. Two examples of its functionality are provided in this section. The first is the optimization of a calibration group that has one subbasin in it, and the second is the optimization of a calibration group that has three subbasins in it.

4.2.1.1 Single Subbasin Calibration Group Optimization

The calibration group associated with the MBLC gage provides a simple example of the effectiveness of the optimization method. This calibration group has only the Yampa 4 subbasin within it because the Yampa 4 subbasin has a gage at its outlet (MBLC) and a gage at its inlet (MBCC). Table 6 shows both the initial model parameters utilized to seed the optimization

method for the Yampa 4 subbasin and the parameters that the optimization method solved. Figure 16 (A) shows the model results utilizing the *initial parameters*. As illustrated by the figure, these model results are inaccurate, and the NSE associated with this model run is -613.05. Once the optimization method was run and optimized parameters were solved, the model was rerun, and the results are shown in Figure 16 (B). As made clear by figure, the optimization solution was able to identify parameters that much more accurately modeled observed gage flows, and the NSE associated with the optimized model run was .89. This is a strong NSE score as it is relatively close to a value of 1 which indicates a perfect model. Furthermore, the Mean Absolute Error (MAE) between the most recent 72 hours of gage data and the model results was 18.2 cfs, which is small in comparison with the flows at the outlet of Yampa 4 that were between 1,850 and 2,100 cfs.

Table 6: Starting parameters used to seed the optimization method and the solution parameters of the optimization method for the Yampa 4 subbasin.

Start Parameters		Optimized Parameters	
Parameter	**Yampa 4**	**Parameter**	**Yampa 4**
Lag Exponent	-0.14	Lag Exponent	-0.22
Scale Coeficient	0.6	Scale Coeficient	3
Scale Exponent	-0.2	Scale Exponent	-0.12

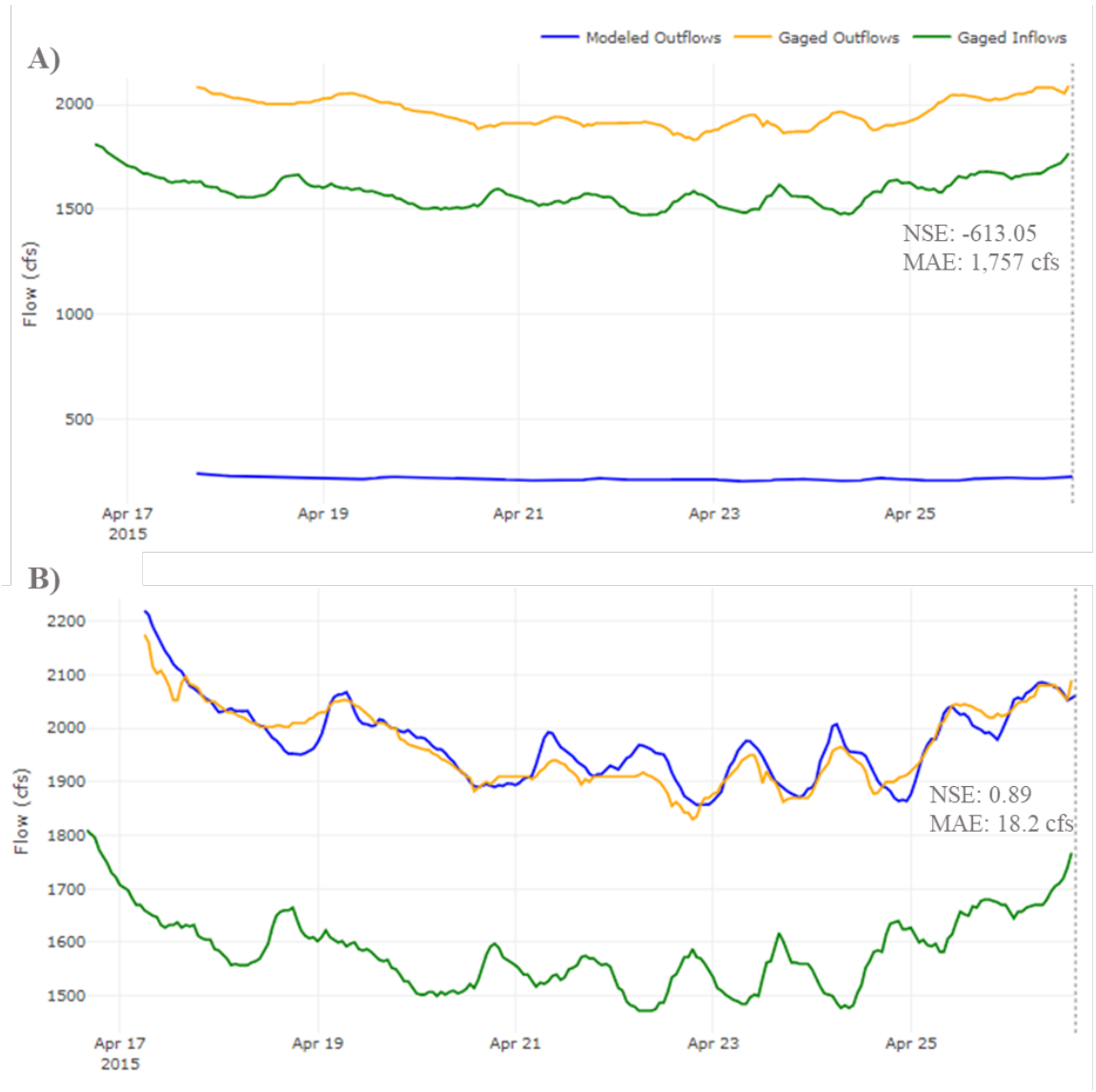

Figure 16: (A) Model results utilizing the initial parameters for Yampa 4 used to seed the optimization method, and (B) Model results utilizing the optimized parameters for Yampa 4 that were solved by the optimization method.

4.2.1.2 Multiple Subbasin Calibration Group Optimization

The calibration group associated with the JESU gage provides a second and more complicated example of the effectiveness of the routing method. The JESU gage is the outlet of the Green 1 subbasin and is the point on the Green River where target flows must be met by reservoir releases from Flaming Gorge Reservoir, and it is important that the model calibrates well with

this gage. The subbasins that calibrate simultaneously to the JESU gage are the Green 1, Green 2, and Yampa 1 subbasins (refer to Figure 1 for the basin map). Table 7 shows both the initial model parameters utilized to seed the optimization method for the calibration group associated with the JESU gage and the parameters that the optimization method solved. Figure 17 (A) shows the model results for the JESU gage utilizing the *initial parameters*. As is shown in the figure, the model results utilizing the initial parameters resulted in consistent low of flows in comparison to the gage. The NSE associated with this model run was -12.08, and the MAE was 452.27 cfs, which is equivalent to model results being biased low between 10-14%. Once the optimization method was run and optimized parameters were solved, the model was rerun, and the results are shown in Figure 17 (B). As is clear in the figure, the optimization solution was able to identify parameters that were much more accurate, and the NSE associated with the optimized model run was .74. This is a strong NSE score as it is well above a value of .5, which is considered a good model. Furthermore, the MAE was for this model run was 53.3 cfs. Given that the flows in the calibration period were between 3,500 cfs and 4,000 cfs, the MAE shows a relatively small error.

Table 7: Starting parameters used to seed the optimization method and the solution parameters of the optimization method for the subbasins in the calibration group associated with the JESU Gage.

Start Parameters			
Parameter	**Yampa 1**	**Green 2**	**Green 1**
Lag Exponent	-0.21	-0.18	-0.25
Scale Coeficient	1	0.99	0.98
Scale Exponent	0.01	-0.01	-0.1

Optimized Parameters			
Parameter	**Yampa 1**	**Green 2**	**Green 1**
Lag Exponent	-0.23	-0.19	-0.22
Scale Coeficient	1.01	1	1
Scale Exponent	0.009	0	0

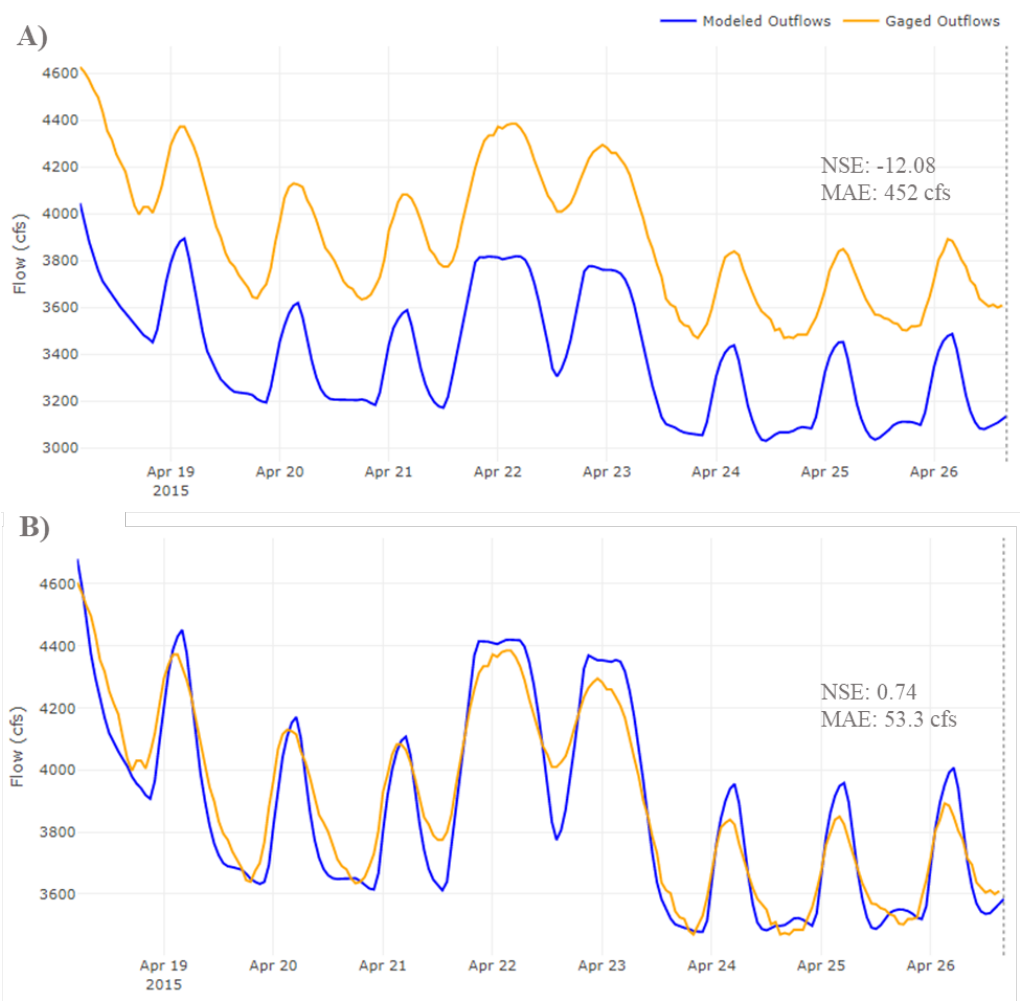

Figure 17: (A) Model results utilizing the initial parameters for the subbasins in the calibration group associated with the JESU gage, and (B) Model results utilizing the optimized parameters for the subbasins in the calibration group associated with the JESU gage.

4.2.2 Optimization Limitations

The limitations of this optimization method are largely due to the complexity that is introduced when optimizing calibration groups that have three subbasins. The following two sections outline two limitations of the optimization method identified during analysis that were due to this complexity.

4.2.2.1 Non-Global Maxima

The first limitation of the optimization method when optimizing a calibration group with three subbasins in it is that it is challenging to be sure that the optimization method found the globally optimal solution. This issue is due to two factors. The first is that when performing optimization on a calibration group with three subbasins, there are nine parameters that must be optimized simultaneously, which corresponds to a nine-dimensional space that is complicated and cannot be visualized. The second factor is that when optimizing a calibration group with three subbasins, the two upstream subbasins must be routed to provide the inflows to the downstream subbasin. Thus, as *parameters* in the upstream subbasins change, the *inputs* to the downstream subbasin change, which makes the optimization of the parameters much more complicated.

Table 8: Two cases of NSE Scores for Initial and Optimized of subbasins in the calibration group associated with the JESU gage. Case 1 was described in more detail in Section 4.2.1.2. Case 2 is a separate example where the optimization was seeded with different initial parameters.

Case 1

Initial Parameters

Parameter	Yampa 1	Green 2	Green 1
Lag Exponent	-0.21	-0.18	-0.25
Scale Coeficient	1	0.99	0.98
Scale Exponent	0.01	-0.01	-0.1

NSE: -12.08

MAE: 452 cfs

Optimized Parameters

Parameter	Yampa 1	Green 2	Green 1
Lag Exponent	-0.23	-0.19	-0.22
Scale Coeficient	1.01	1	1
Scale Exponent	0.009	0	0

NSE 0.74

MAE: 53.3 cfs

Case 2

Initial Parameters

Parameter	Yampa 1	Green 2	Green 1
Lag Exponent	-0.2	-0.2	-0.2
Scale Coeficient	1.02	1.02	1.02
Scale Exponent	0.01	0.01	0.01

NSE: -17.22

MAE: 575 cfs

Optimized Parameters

Parameter	Yampa 1	Green 2	Green 1
Lag Exponent	-0.24	-0.22	-0.2
Scale Coeficient	1.02	0.98	1.02
Scale Exponent	0.01	-0.01	0

NSE -2.13

MAE: 161 cfs

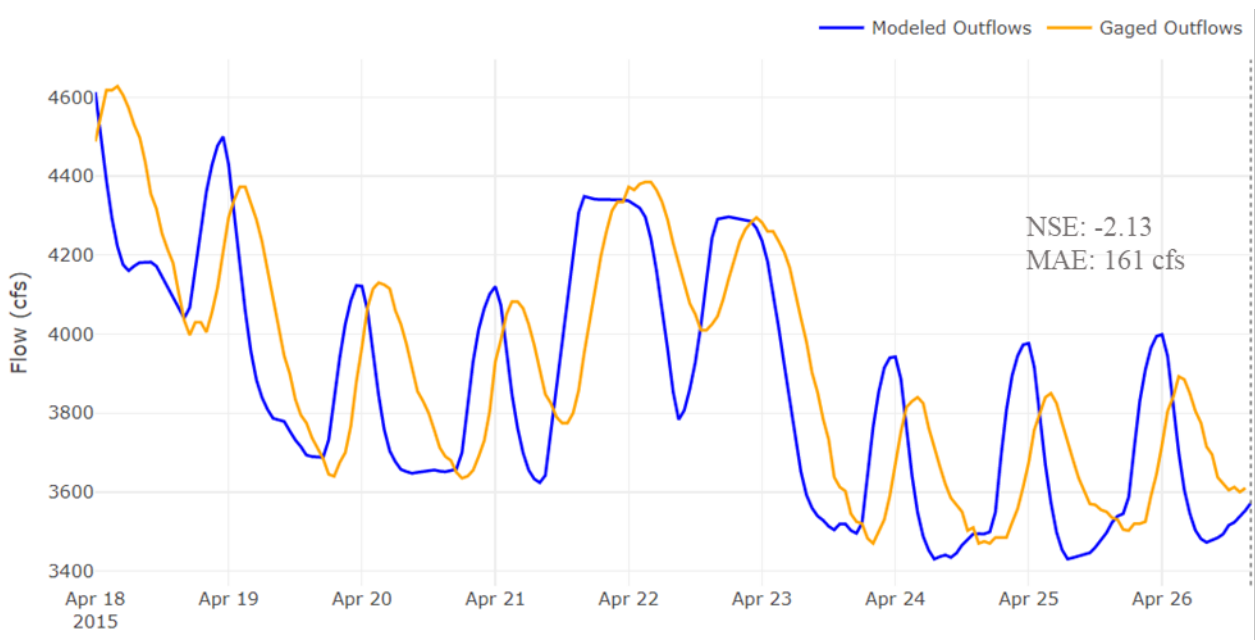

Figure 18: Model results at the JESU Gage utilizing the optimized parameters of Case 2 in Table 8.

The calibration group associated with the JESU gage provides a good example of this limitation. Table 8 provides a summary of optimizations that were performed with two sets of different initial parameters. Case 1 is the same case that was described in Section 4.2.1.2, and the NSE was improved from -12.08 to 0.74. In Case 2, the optimization was seeded with slightly different parameters than in Case 1. The NSE associated with initial parameters in Case 2 was -17.22, and the optimization method found a local maximum of NSE of -2.13, at which point it could no longer improved the model. Figure 18, shows the modelled gage flows at the JESU Gage utilizing the optimized parameters from Case 2. From the figure, the most notable issue with the optimized model results is that the modelled flows are shifted in time from the actual gage flows. Moreover, the optimized lag coefficient parameters in Case 1, which performed well, are considerably different than those of Case 2, which performed poorly. This is evidence that the initial parameters used to seed the optimization method can affect the performance of the optimization, and it is challenging to know if the optimization method has found the global maximum. Therefore, in practice, it would be the user's discretion to decide what is an adequate model performance to consider the model usable.

4.2.2.2 Overfitting

A second limitation surrounding the optimization method with calibration groups with three subbasins in it is that the two upstream subbasins that provide inflows for the downstream subbasin are ungagged, and the optimization method can manipulate the parameters of the upstream subbasins in any such way within the boundary of the given parameter constraints. If not monitored, this can result in unrealistic flows in one or both upstream subbasins. An example of this occurs in the calibration group associated with the YNHC Gage, which includes the Elk 1, Yampa 7, and Yampa 6 subbasins. The optimization parameters associated with this calibration

group results in a NSE model performance of 0.99, which is extremely high. Figure 19, below, shows the model results of the optimization solution for the three subbasins in the calibration group. The model results show the Elk 1 subbasin, which represents only a three-mile section of the Elk River, consistently loosing roughly 80 cfs of flow through the reach. Furthermore, the optimization solution shows the Yampa 7 subbasin gaining around 150 cfs consistently. It is highly unlikely that the gains and losses in these sections of river behave like this, and the more likely issue is that the optimization method is over fitting by adjusting the flows in both the Elk 1 and Yampa 7 subbasins in such a way that provides inflows to the Yampa 6 subbasin that can be routed to fit the YNHC Gage *very* well.

The inherent challenge with this problem is that the outlets of the Elk 1 and Yampa 7 subbasins do not have a gage, so it is impossible to know exactly how much flow is occurring at these points. It is important in this case to have knowledge about the river and what is happening on the river in these sections to better estimate what the flows might be, even if it comes at the expense of model performance. For example, the reach of the Yampa River contained in the Yampa 7 subbasin flows through Steamboat Springs, Colorado, a mountainous region with many small streams that flow into it during the snow melt season. Furthermore, this stretch of the Yampa River is a much longer reach of river than the reach of the Elk River associated with the Elk 1 subbasin. The more likely scenario is that the flows through the Elk 1 subbasin are remaining relatively constant because it represents only a 3-mile section of river, and the flows through the Yampa 7 subbasin are gaining. To address this issue, a manual calibration was performed on the optimization results by adjusting the Scale Exponent on both the Elk 1 and Yampa 7 reaches. The model results for the manually calibrated parameters are shown in Figure 20. While these adjustments resulted in a NSE model performance of .94 which less than that of

the optimization solution, these parameters produce more realistic results. As a response to this issue, the constraints on the optimization could be updated to force the optimized solution for the parameters to vary within a more realistic range, thus improving the model.

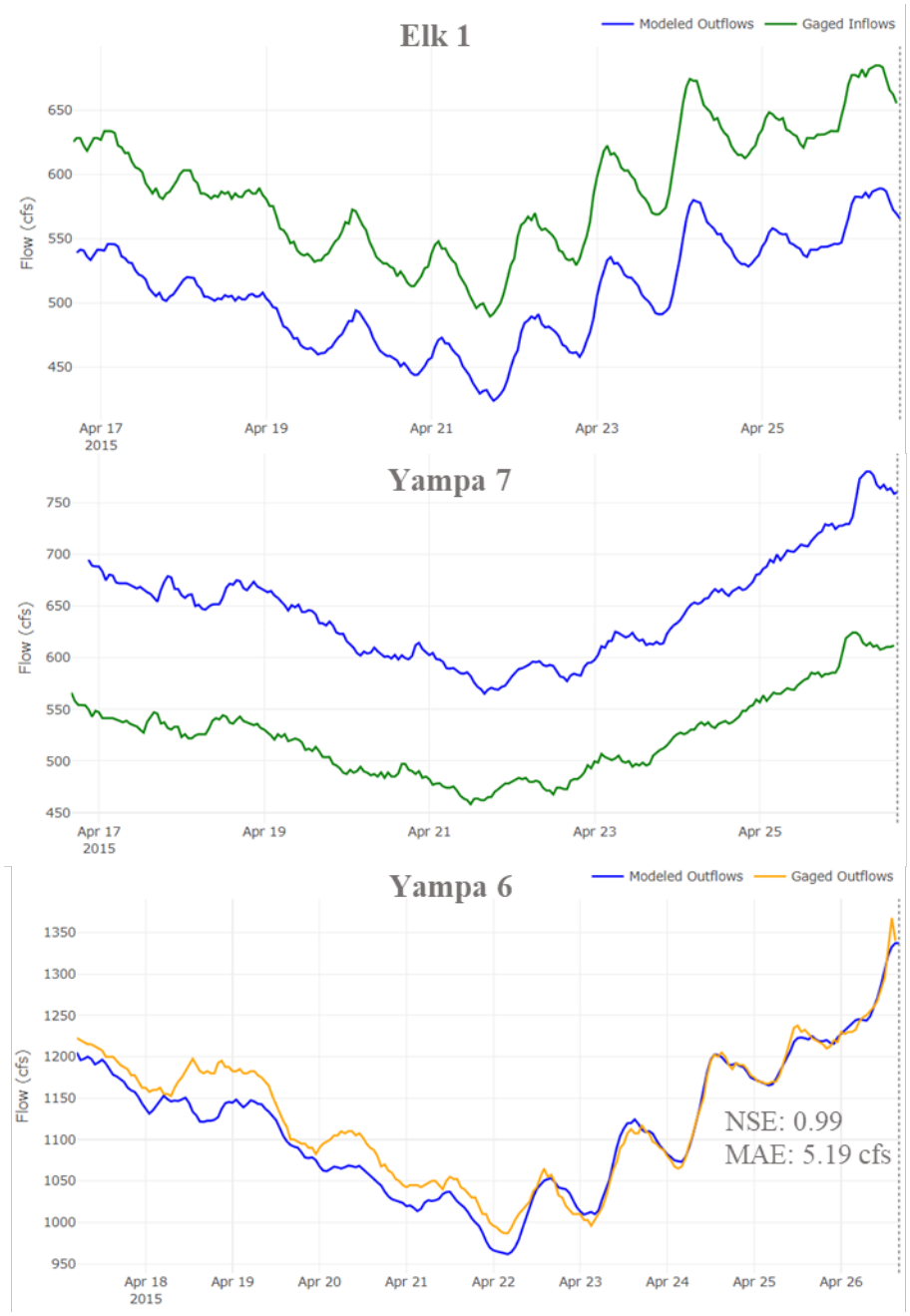

Figure 19: Model results utilizing optimization parameters of outflows of the Elk 1 (top), Yampa 7 (middle), and Yampa 6 (bottom) subbasins.

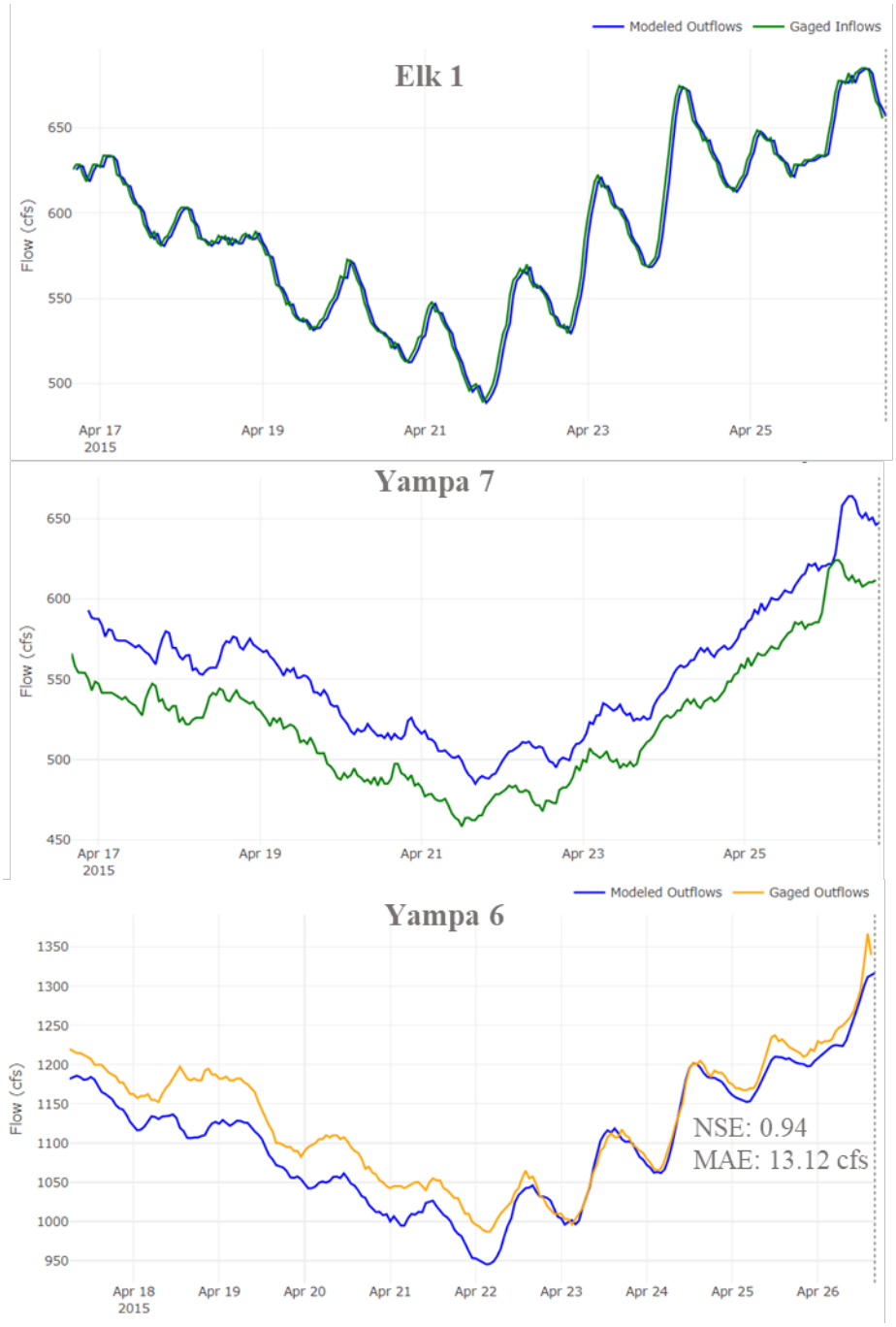

Figure 20: Model results utilizing parameters that were manually calibrated of the Elk 1 (top), Yampa 7 (middle), and Yampa 6 (bottom) subbasins.

4.3 Reservoir Scheduling Analysis

This section provides analysis of the prototype model's ability to facilitate the scheduling of reservoir releases to meet the downstream target at the JESU gage. Prior to scheduling reservoir releases, the entire model needed to be calibrated to the observed data. Table 9 provides a summary of model calibration by calibration gages. The worst performing calibration within the model was the LILC calibration gage, which has an NSE of 0.4. While this is not a very strong NSE, the MAE was only 18.5 cfs, which suggests that the flows on this section of the river represent a small portion of the flows that are being routed to the JESU gage target location on the river and will have less bearing on the overall error in the model. The best performing calibration gage was the YPDC gage, which had an NSE of 0.95 and an MAE of 10.4 cfs. Lastly, the JESU gage scored an NSE of 0.74 and an MAE of 53.3 cfs. While this is a larger MAE, the flows on the river in the calibration period were between 3,500 and 4,000 cfs which is large in comparison to the MAE.

Table 9: Performance scores of the prototype model at each calibration gage. The model calibrated best to the YPDC gage and calibrated the worst to the LILC gage.

Calibration Gage	Calibration Group Subbasins	NSE	MAE (cfs)
LILC	Slater Fork 1, Little Snake 3, Little Snake 2	0.4	18.5
YNHC	Elk 1, Yampa 7, Yampa 6	0.94	13.1
YBCC	Yampa 5	0.85	16.38
MBLC	Yampa 4	0.89	18.47
YPDC	Little Snake 1, Yampa 3, Yampa 2	0.95	10.4
JESU	Yampa 1, Green 2, Green 1	0.74	53.3

Once the model was calibrated, both an hourly time series target flow at the JESU gage and an hourly time series reservoir release schedule from Flaming Gorge Reservoir was created as a

test. In creating the initial reservoir releases schedule, the number of release changes allowed per day was limited to two to more accurately model real-world scenarios in which operators try to limit the number of release changes that are made. Figure 21 provides a plot of the release schedule from Flaming Gorge Reservoir, the flow targets at the JESU gage, and the model results for flow at the JESU Gage given the reservoir release schedule. Given the input target flows at the JESU gage, the input reservoir release schedule did not meet the targets at the JESU gage during the following time periods:

- From 8 am to 9 pm on April 28th.
- From 3 am to 6 pm on April 29th.
- From 1 am to 4 pm on April 30th.
- From 11 pm on April 30th to 9 am on May 2nd.

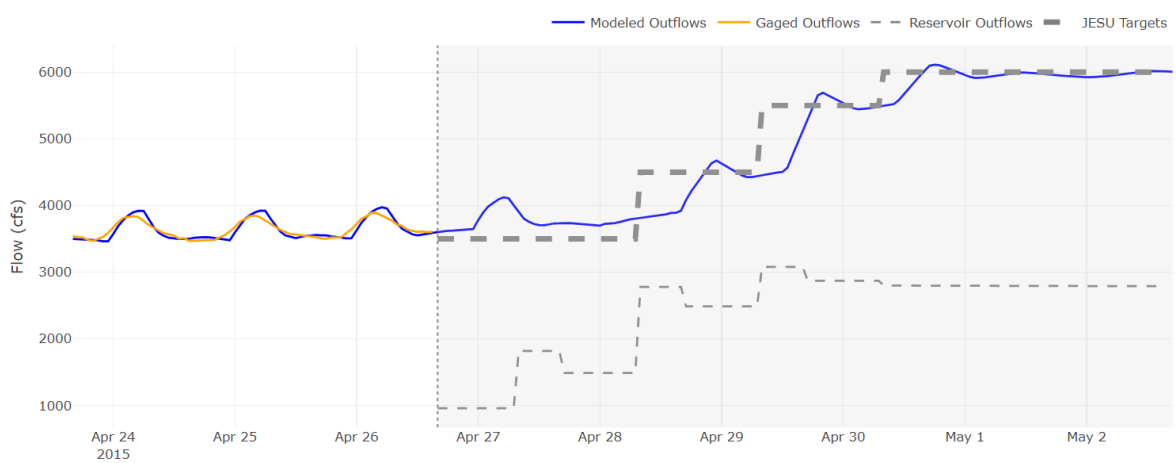

Figure 21: Modelled Future Flows at the JESU target gage given observed and scheduled releases of Flaming Gorge Reservoir. The darker shaded region on the plot represents the future time-period at the JESU gage, and the thick dashed line represents the targets at the JESU gage.

A revised reservoir schedule was created to better meet the targets. The revised schedule limited to one to two releases changes per day that occurred no earlier than 8 am and no later than 8 pm.

The results of the revised schedule are shown in Figure 22. As illustrated in the figure, the revised schedule meets the downstream flow targets at every timestep in the future. It is worth noting that there are short periods where large excess reservoir releases are being made. For example, during the early hours of April 28th, the river begins to gain flow several hours before the target is raised to 4,500 cfs. This could be better optimized so as not to increase reservoir releases too early; however, this would require a change in release in the middle of the night, which is not realistic given that most reservoirs have release mechanisms that are manually operated.

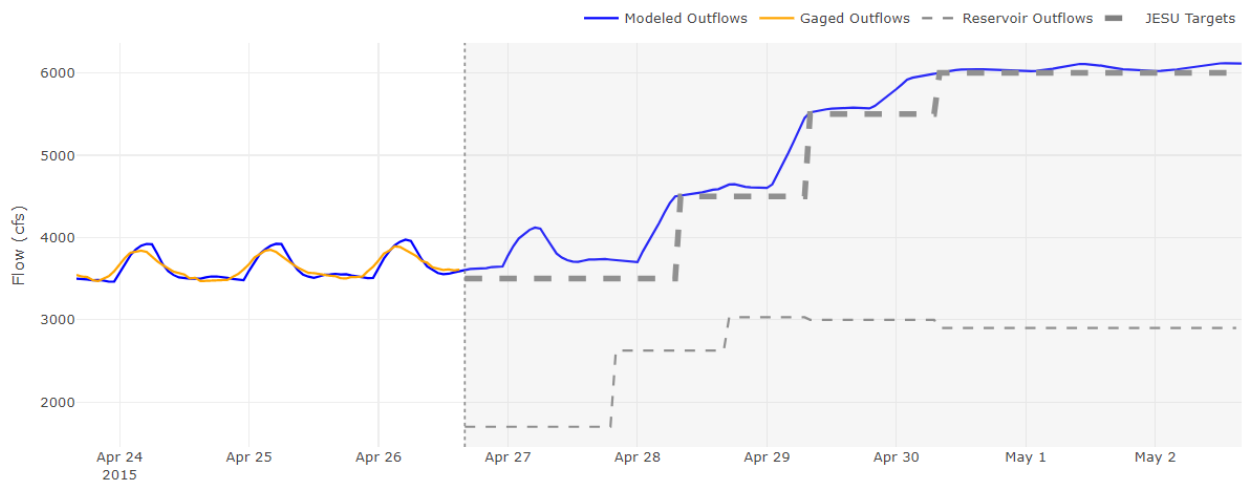

Figure 22: Modelled Future Flows at the JESU target gage given observed and a modified release schedule of Flaming Gorge Reservoir to avoid missing target flows downstream. The darker shaded region on the plot represents the future time-period at the JESU gage, and the thick dashed line represents the targets at the JESU gage.

5. Future Development

As is typical with almost any piece of software development, there are many ways the prototype model could be developed to increases its functionality and allow it to be used in practice. This

report identifies three of the most important pieces of development that would enhance the prototype model to a point that it could be utilized effectively in the real world.

The first and more essential pieces of development would be to build the functionality to update the gage data in the model to be current. This would allow for the ability to utilize the model in real time. As the prototype model was coded with this development in mind, the process for this development would be relatively simple, and it would involve setting up regular scripts that pull data from USGS and then update the data within the model database. This script could be automated so that the user never has to worry about data updating. The only thing that would need to be incorporated in this process is data quality control. Processes could be set up to automatically handle most data problems, but it is highly likely that algorithms would *always* be able to fix data issues correctly. Thus, providing a way for the user to make changes to the gage data where there were bad/missing measurements would be necessary.

A second piece of future development that would enhance the capabilities of this model as a real-world application would be to develop an interface that allows the user to build and manipulate forecasts. This would involve querying hourly forecasts of input subbasins in the model that are produced daily by the Colorado Basin River Forecasting Center (CBRFC) and storing them in the model database, and then developing an interface for the user to make alterations to these forecasts that can be utilized by the model to answer questions about potential futures.

A third piece of future development would be enhanced plotting capabilities in the model that allow the user to visually see when water that has been routed is the result of gaged inputs, or the result of forecasted inputs. This would allow the user to assess model results more adequately by knowing when routed data originated from forecasts, which are inherently more uncertain than gaged data.

6. Conclusion

The objective to evaluate the feasibility of using a web-based application to schedule reservoir releases to meet a downstream target was met was met by the prototype model as described in this report. It was shown that the river routing method utilized by the prototype model performs well as compared to more established methods. Furthermore, the optimization method utilized by the prototype model also performed efficently: in many cases the optimization method was able to find the global solution. However, as is common with optimization, it is still important that when optimizing the model users have a good understanding of the river to avoid over fitting and producing unrealistic model results. Lastly, it was shown that the prototype model was an effective tool at scheduling reservoir releases to meet a downstream target.

To generalize the idea of "feasibility" in water resources modelling through web-based applications, this prototype model showed that the web-based application framework allows a developer to incorporate complicated pieces of open-source technology, such as the SciPy Python Library, into water resources modeling to accomplish complex mathematical operations. This was exhibited by implementation of the Optimization Method in the Prototype Model. Furthermore, the web-based application framework allows full control over the way the user can interact with the model. The user interface of the prototype model, built for a marginally technical user with a good understanding of the modeled river basin, was developed resolutely to prevent the user from being able to break the model while using it. The user can only interact with the model via well-defined actions and visuals provided by user interface. It prevents the user from being able to see information that is unnecessary for them to see and to use the model efficiently.

In conclusion, the development of engineering tools in a web-based interface in the field of water resources management such as the prototype model described in this report would greatly enhance the computational ability, access, and usability of modelling technologies. While it is not a substitute for understanding the physical characteristics or institutional structures of an area of application, it would enhance the usability of complex models, providing a more efficient means of delivering answers to challenging and complicated problems.

7. References

Applying the Muskingum Routing Method. (2021, September 19). Retrieved from HEC-HMS Tutorials and Guides: https://www.hec.usace.army.mil/confluence/hmsdocs/hmsguides/applying-reach-routing-methods-within-hec-hms/applying-the-muskingum-routing-method

Coors, S. (2003). *User Manual.*

Coors, S. (2021, January 15). Routing Method Discussion. (P. Noe, Interviewer)

django. (2021, February 15). Retrieved from django: https://www.djangoproject.com/

django REST framework. (2021, February 15). Retrieved from django REST framework: https://www.django-rest-framework.org/

Google Earth. (2021, January 15). Retrieved from Google: https://earth.google.com/web/

Nation Water Information System: Mapper. (2021, March 15). Retrieved from United States Geological Survey: https://waterdata.usgs.gov/nwis

Niemann, J. (2020). *Measures of Model Performance, CIVE 524.* Fort Collins, Colorado.

Numpy v1.21 Manual. (2021, January 10). Retrieved from NumPy: https://numpy.org/doc/stable/

python. (2020, November 20). Retrieved from python.org: https://www.python.org/

SciPy.org. (2021, March 15). Retrieved from SciPy.org: https://www.scipy.org/

Vue.js. (2021, April 22). Retrieved from Vue.js: https://vuejs.org/

Yang, X.-S. (2017, March 15). *Engineering Mathematics with Examples and Applications.* London, United Kingdom: Academic Press. Retrieved from Science Direct: https://www.sciencedirect.com/topics/mathematics/sequential-quadratic-programming