NATIONAL WEATHER SERVICE
OFFICE of HYDROLOGIC DEVELOPMENT

TECHNICAL DESIGN

Reservoir Tools Enhancement Project

Version 4.3

Revision History

Date	Version	Description	Author
05/09/2007	4.1	Initial Draft	D. Sharp (RTi)
05/23/2007	4.2	Deliverable version for OHD review	D. Sharp (RTi)
09/07/2007	4.3	Incorporate OHD revisions and Final Changes	D. Sharp (RTi) J Gofus (OHD) K. Hsu (OHD)

Table of Contents

1. Introduction

The National Weather Service River Forecast System (NWSRFS) is used by the National Weather Service (NWS) to generate river and flash flood forecasts. The Joint Reservoir Regulation operation (RES-J) of NWSRFS is used to model a single reservoir or a system of reservoirs with associated river reaches. RES-J solves a network of reservoirs, reaches, and nodes from upstream to downstream, simulating conditions including inflows, releases, withdrawals, pool elevations, and in-stream discharges.

Operational use and testing of the NWSRFS and RES-J, including their applications to streamflow regulation accounting at the Missouri Basin River Forecast Center (MBRFC), has revealed the need and value of enhancing NWSRFS and RES-J functionality.

The following two enhancements were identified for inclusion in this task:

1. Multi-valued Time Series Lookup Operation – The existing LOOKUP3 operation linearly interpolates for a dependent result given two independent arguments and a family of curves relating the two independent arguments to the dependent argument. This method utilizes a 3-variable relationship between upstream flow (Q1), downstream elevation (E2) or flow (Q2) and elevation from the upstream location (E1): E1 = f (Q1, E2). Only single valued time series are allowed. Soil moisture zone contents and runoff time series can be useful in making operational decisions related to streamflow regulation. For example, soil moisture and runoff can be considered when estimating diversions that are occurring for agriculture. The Sacramento model includes two multi-value time series containing zone contents and runoff components. The values from these time series may be plotted using PLOT-TS (e.g., to facilitate calibration); however, the time series cannot otherwise be accessed for manipulation by other NWSRFS operations. Enhancing the LOOKUP3 operation to have access to the multi-value time series will allow the information to be used in regulation modeling. Additionally, allowing a lookup based on the day and time within a year will allow operating rules to be specified in lookup tables.
2. Integrate NWSRFS Rating Curves – The existing MAXSTAGE method in RES-J allows control of a reservoir release based only on allowable stage at a downstream streamflow station control point. The MAXSTAGE method requires the user to input a rating curve for the downstream node. It is requested that rating curves defined in the system be accessible to this method in place of the rating curve defined in RES-J. Additionally, the rating curve will always be entered through the NODE Component corresponding to the downstream station. Adding the rating curve at nodes will allow the specific nodes to be used as a forecast point. An option will be allowed to enter a rating table at any node in lieu of the rating curve. The existing rating table in the MAXSTAGE Method will be retained only for backward compatibility. This enhancement will ensure the model always uses the most up-to-date rating curve data from the NWSRFS database. The current MAXSTAGE method also requires that a stage value be specified, which limits application in cases where operations are based on discharge; consequently, the ability to specify a maximum discharge is needed.

The enhanced functionality will be available in the following NWSRFS programs: the Forecast Component Initialization Program (FCINIT), the Forecast Program (FCST, including Forecast Execute – FCEXEC), the Interactive Forecast Program (IFP), the Manual Calibration Program (MCP3) and the Automated Optimization Program (OPT3).

This design document describes the enhancements in preparation for their development. The audience is expected to be largely familiar with the NWSRFS and the RES-J operation – its overall functionality and descriptive terms. As necessary, the audience is encouraged to reference existent NWSRFS and RES-J documentation for further background and certain details described in this document.

Specifically, this Technical Design Document focuses on:

1. Input specifications
2. Output specifications
3. Changes to source code

2. Design for LOOKUP3 Operation

2.1 Current State of Operations

It is useful to utilize a lookup table to implement operational decisions, for example, estimating a diversion or reservoir operation based on a rule curve or physical condition (e.g., soil moisture). The NWSRFS LOOKUP3 operation provides existing capabilities to lookup a dependant variable based on two independent variables. However, the existing software does not meet the requirements noted in the CONOPS, in particular to utilize multi-value time series as input and allow using a time step date as one of the independent variables in the lookup. Currently, only single-value time series can be used for the LOOKUP3 operation, and both independent variables must be values in time series.

2.2 Functional Solution Description

Adding multi-value time series lookup capability to the NWSRFS operation will provide the ability for a modeler to generate a new time series by utilizing a three-variable lookup table with one or both independent variables being multi-value time series.

2.3 Inputs

The following section explains the parametric information required for the enhanced LOOKUP3 method. It specifies the relevant section of the NWSRFS User Manual Documentation that will change to reflect the enhanced LOOKUP3 capabilities. Changes to the existing document are denoted by highlights.

2.3.1 V.3.3-LOOKUP3 3 VARIABLE TABLE LOOKUP

Input Summary: The card input for this Operation is read in free format and is as follows:

Card	Field	Format	Contents
1	1	A72	General information for this operation
2	1	A8	1st independent argument (X) time series identifier
	2	A4	1st independent argument (X) data type code
	3	I	1st independent argument (X) data time interval (hours)
	4	A8	1st independent argument (X) multi-value time series data type;

			only if the 1st independent argument is a multi-value time series (see below)
3	1	A8	2nd independent argument (Z) time series identifier
	2	A4	2nd independent argument (Z) data type code
	3	I	2nd independent argument (Z) data time interval (hours)
	4	A8	2nd independent argument (Z) multi-value time series data type; only if the 2nd independent argument is a multi-value time series (see below)

Explanation of Multi-Value Time Series Data Type:

Depending on the data type of the time series data type (the only valid types are ROCL and SMZC) this field must be one of the following Multi-Value Data Types. See NWSRFS User Manual Parts II.4-LOOKUP3, and V.3.3-LOOKUP3 for additional details

Time Series Data Type	Multi-Value Data Type	Description Value (Order in Time Series)
SMZC	UZTDEF	Upper zone tension water deficit (1)
	UZFWC	Upper zone free water contents (2)
	LZTDEF	Lower zone tension water deficit (3)
	LZFSC	Lower zone free supplemental contents (4)
	LZFPC	Lower zone free primary contents (5)
ROCL	TCHANINF	Total channel inflow (1)
	IMP-RO	Impervious runoff (2)
	DIR-RO	Direct runoff (3)
	SUR-RO	Surface runoff (4)
	INTERFLO	Interflow (5)
	SUPBASE	Supplemental baseflow (6)

	PRIMBASE	Primary baseflow (7)

For SMZC and ROCL the Multi-Value Data Type specifies which element of the multi-value time series is selected. For instance, with a Time Series Data Type of SMZC and a Multi-Value Data Type of UZFWC, the 2nd element (of 5) in a particular time step is specified as the element of interest. (See NWSRFS Part V.3.3-SAC-SMA for additional details). This is an optional parameter/field. If no value is specified for the fourth field, the Time Series data are assumed to be one value per time step (i.e. no change from the current behavior).

If the Multi-Value Data Type for a Time Series which supports Multi-Value Data Types (i.e. ROCL or SMZC) is blank, the Multi-Value Data Type is assumed to default to the first element in the Data Type array (UZTDEF for SMZC; TCHANINF for ROCL).

2.4 Outputs

The following section explains the outputs from the enhanced LOOKUP3 operation. Unless noted below, all outputs remain the same as the original operation (i.e. there is no output from the execution routine).

Error and Warning Messages: The error and warning messages generated by this operation and the corrective action to take when they occur are as follows:

14. ** LOOKUP3 INPUT ERROR**

    A multi-value time series identifier (card 2 or 3, field 1) has been specified, but no multi-value time series data type (card 2 or 3, field 4) has been specified. Field 4 is a required parameter for multi-valued time series. Time series identifier = XXX.

    Action: Check time series identifies.

15. ** LOOKUP3 INPUT ERROR**

    Invalid Time Series and Time Series Data Type Pair. Time Series = XXX; Time Series Data Type = YYY.

    Action: Check time series type and multivalue index.

2.5 Source Code

ex53.f, prp53.f, pin53.f, puc53.f, chekmv.f, cvtidx.f, extrts.f

2.6 Detailed Design

The NWSRFS Operation Code for the LOOKUP3 Operation is 53. Following are the changes required to each affected source code file.

2.6.1 pin53.f

This is the ParameterINput routine. It inputs all cards from the input deck and populates the Parameter (NWSRFS P) array. The P array is defined as follows:

POSITION	CONTENTS OF P ARRAY
1	VERSION NUMBER OF OPERATION
2-19	DESCRIPTION - TITLE
20	# OF POINTS IN THE P ARRAY
21-22	1ST INDEPENDENT VARIABLE (X) TIME SERIES IDENTIFIER
23	1ST INDEPENDENT VARIABLE (X) TIME SERIES DATA TYPE CODE
24	1ST INDEPENDENT VARIABLE (X) TIME SERIES TIME INTERVAL
25-26	2ND INDEPENDENT VARIABLE (Z) TIME SERIES IDENTIFIER
27	2ND INDEPENDENT VARIABLE (Z) TIME SERIES DATA TYPE CODE
28	2ND INDEPENDENT VARIABLE (Z) TIME SERIES TIME INTERVAL
29-30	RESULTANT (Y) TIME SERIES IDENTIFIER
31	RESULTANT (Y) TIME SERIES DATA TYPE CODE
32	RESULTANT (Y) TIME SERIES TIME INTERVAL
33	THE UNITS THAT THE USER ENTERS THE Z SEGMENT ARRAY 'ENGL' OR 'METR' (DEFAULT)
34	NUMBER OF POINTS IN THE Z SEGMENT ARRAY
	Z SEGMENTS (Z1,X11,Y11,X12,Y12,Z2,X21,Y21,X22,Y22,ETC)
35+	Z SEGMENT ARRAY

Changes Required:

1. Declare a local char variable CMVIDX (CharacterMultiValueInDeX) and a local int variable IMVIDX (IntegerMultiValueInDeX) to hold the multi-value time series data type used to index into the multi-value time series.
2. For each input time series do:
   i. Read in the CMVIDX field from the card, if it exists
   ii. If CMVIDX is non-null:
      - Check the validity of the CMVIDX variable against the time series type; print error message and exit if invalid
      - If valid, convert CMVIDX to an integer (IMVIDX)
      - Store IMVIDX in the decimal portion of the input time series time interval (P(24) or P(28); in order to continue using existing P array as is); Note that the P(24) and P(28) elements are 4 characters wide this allows for one decimal place to hold the index number (i.e., the index number must be 0-9 – only up to 9 values per time step are supported)
      - If the multi-value data type is null (for a valid multi-value time series), assign the multi-value data type to be the first element of the multi-value data type array

   iii. If CMVIDX is null, no changes are required

2.6.2 ex53.f

This is the execute routine. It executes the LOOKUP3 operation and returns the interpolated dependent variable as a function of two independent input variables (time series).

Changes Required:

1. Declare local int IMVIDX as an index into the input arrays
2. Compute the elements of the input (D1X and D2Z) and output (SY) time series to process using the following equations

   Start = ((IDA-IDADAT)*(24/IDT))+((IHR-1)/(IDT*NVPDT)+IMVIDX

   Finish = ((LDA-IDADAT)*(24/IDT))+((LHR-1)/(IDT*NVPDT)+IMVIDX

   Where:

   - IDA = Julian date of first day to compute
   - LDA = Julian date of last day to compute
   - IDADAT = Julian date of 1st day of time series data
   - IDT = Time step in hours
   - IHR = First hour to compute
   - LHR = Last hour to compute
   - NVPDT = Number values per time step (defaults to 1); determined via a call to FDCODE
   - IMVIDX = Index into the values in a time step (defaults to 1); extracted from P(24) (D1X) or P(28) (D2Z) when applicable

   For each time series, define the following:

   - BD1X = beginning element of D1X
   - ED1X = ending element of D1X
   - BD2Z = beginning element of D2Z
   - ED2Z = ending element of D2Z
   - BSY = beginning element of SY
   - ESY = ending element of SY
   - STEPX = NVPDT for D1X
   - STEPZ = NVPDT for D2Z

3. Process the time series using the following algorithm:

```
// initialize the indices into the time series arrays
```

```
IDXZ=BD2Z
// for each time step
DO IDXY = BSY to ESY
   // determine the X value to use in the lookup
  X = D1X(IDXX)
  Z = D2Z(IDXZ)
  // TABLE is the TABLE from the P array,
  // i.e. the Z segments
  Call LOOKUPTABLE(X,Z,TABLE,Y)
  SY(IDXY) = Y
  IDXX += STEPX ENDIF
  IDXZ += STEPZ
ENDDO
```

2.6.3 prp53.f

This routine reformats (for readability) and prints out the data the operator entered via the input deck.

1. Extract number of values per time series (from the P array, element 24 or 28, decimal portion)
2. If number of values per time step > 1 then add number of values per time step to the output

2.6.4 puc53.f

This routine punches the data defined in the FS5 files in a format suitable for re-entry with no reformatting required.

Changes required:

1. Extract number of values per time series (from the P array, element 24 or 28, decimal portion)
2. If number of values per time step > 1 then add number of values per time step to the output

2.6.5 chekmv.f, cvtidx.f, extrts.f

These are all new utility files to respectively:

- check the validity of multivalue time series’
- convert between multivalue data types (alpha) and numerical index values
- extract time series from an input card

3. Design for NODE Component RES-J Operation

3.1 Current State of Operations

The RES-J MAXSTAGE method iteratively solves for the release of a reservoir until the downstream stage (corresponding to the routed release plus any flows originating downstream of, or parallel with, the reservoir) at a control point (node) is within a specified tolerance of a specified maximum value.

Currently, the RATING_CURVE (stage elevation / discharge pairs) for the node is specified in the MAXSTAGE method.

3.2 Functional Solution Description

Specification of the rating curve/table information will be moved from the RES-J MAXSTAGE method to the controlling node. Storing the rating curve/table information at the node will provide more physically appropriate definition of data because the information is specific to the node. Additionally, the capability to access a rating curve from the operational data will be incorporated (with the access occurring through specification of the rating curve ID at the node). This allows the current data in an operational system to be accessed without defining a table, and assures any forecast will use the most up-todate database data in an operational system.

3.3 Input

The following section explains the parametric information required for the enhanced RES-J MAXSTAGE method. It specifies the relevant section of the NWSRFS User Manual Documentation that will change to reflect the enhanced RES-J MAXSTAGE capabilities. Changes to the existing document are denoted by highlights.

V.3.3-RES-J JOINT RESERVOIR REGULATION OPERATION

Parameters for each NODE Component

A node is used as a confluence point of flows and as a checkpoint for certain stage / discharge relationships found in Method MAXSTAGE.

Because a node is a point in the system, no routing or other transformation of inflow to outflow occurs; what comes in immediately goes out.

Keyword	Definition and Format
NODE	Indicates the start of node parametric information; 'NODE' must be followed by a node-type identifier defined in the topology section.

[CONSTANT]

Indicates specification of a constant to be used in the RULES definition section. These constants should not be used more than once in an English unit parameterization. Otherwise the conversion to RES-J internal units (SI units) will be erroneously performed more than once. 'CONSTANT' must be followed by:

- 'component identifier.constant identifier'
- the constant’s value

TSINPUT

Indicates the existence of an inflow time series to be applied at the top of the reach. 'TSINPUT' must be followed by:

- time series keyword 'INFLOW'
- an input time series alias defined in the TIMESERIES definition section 1/

While there is no foreseeable reason to place a node at the top of the system topology tree (no components further upstream) any node so positioned must have at least one usage of 'TSINPUT'. Nodes elsewhere in the system may use 'TSINPUT' to define a local inflow, if desired, with outflow from another component acting as the primary (and necessary) inflow to the reach. Outflow from upstream components must not be defined using TSINPUT. Multiple uses of TSINPUT are acceptable.

[TSOUTPUT]

Indicates that one of the node state time series will be output from RES-J. 'TSOUTPUT' must be followed by:

- time series keyword 'OUTFLOW’
- an output time series alias defined in the TIMESERIES definition section 1/

multiple uses usage of TSOUTPUT are acceptable

[DISCHARGE]

Initial discharge at the node; value must follow 4/

[PREVIOUSDISCHARGE]

Discharge at the node one step prior to carryover save; value must follow 4/

[RATINGCURVEID]

NWSRFS Rating Curve ID; value must follow (a rating curve id

	can not be specified if a TABLE is also defined, see below); a valid Rating Curve ID is <= 8 characters long
[TABLE]	Indicates the start of the stage/discharge rating curve table. TABLE must be followed by the 'RATING_CURVE' keyword. Beginning on the next line after 'RATING_CURVE' create a table of stage elevation / discharge pairs describing the channel hydraulic conditions at the node. A rating curve TABLE cannot be specified if a RATINGCURVEID is specified.
[ENDTABLE]	Indicates the end of the rating curve table.
ENDNODE	Indicates the end of node parametric information

3.4 Error Messages

Error messages will be generated for the following error conditions:

- A rating table is specified at a node, as well as a rating curve id (CONOPS 4.2.3)

** NODE DEFINITION INPUT ERROR**

A rating table and a rating curve ID are specified at a single node (NODE_ID).

Action: Check NODE_ID node definition.

3.5 Source Code

Node_SetGet.cxx, Node_Constructors.cxx, Node_initialize.cxx, Node.h

3.6 Detailed Design

As much as possible, the implementation for the MAXSTAGE enhancements will make use of existing code – especially code for STAGE-Q. Existing STAGE-Q code already implements a number of the requirements (linear and logarithmic interpolation of rating curves, for instance), as well as providing entry points for additional functionality that may be desired in the future (e.g., looped rating curves).

To this end the STAGE-Q FSTQC routine will be used as the primary "engine" for conversion between stage/discharge (and vice versa) when a rating curve ID is specified. When a rating TABLE is specified as input, an existing MaxStage routine that does a simple lookup and interpolation will be used. In both of these cases, the ultimate goal is to calculate a maximum flow that can be used in the existing MaxStage algorithm. After the maximum flow is calculated, no changes will need to be made to the MaxStage algorithm (except to allow for specification of a MaximumDischarge instead of a MaximumFlow).

Note that if a rating curve id is defined at the node, the relevant parameters (NCROS, FRLOOP, EMPTY, etc) from the FRATNG Common Block will be used by STAGE-Q during the conversion.

The following source code files are involved in the implementation of the MAXSTAGE enhancements.

3.6.1 Node_setGet.cxx

Routine to read in the node parameterization/definitions.

Changes Required:

- Add the capability to recognize new (optional) keywords RATINGCURVEID and TABLE
- If RATINGCURVEID is defined, read in the ID and place in the NODE object
- If TABLE is defined, read in the table and place in the NODE object
- If both RATINGCURVEID and TABLE are defined, print error message

3.6.2 Node_Constructors.cxx

C++ Constructors for a new node object.

Changes Required:

- Add capability to account for new members of the node class

3.6.3 Node_initialize.cxx

Initialize a new Node.

Changes Required:

- Initialize the new members of the node class

3.6.4 Node.h

Definitions for the Node class.

Changes required:

- Add _rating_curve_id and _rating_table members to the Node class definition

4. Design for MAXSTAGE Method RES-J Operation

4.1 Current State of Operations

The RES-J MAXSTAGE method iteratively solves for the release of a reservoir until the downstream stage (corresponding to the routed release plus any flows originating downstream of, or parallel with, the reservoir) at a control point (node) is within a specified tolerance of a specified maximum value.

Currently, the RATING_CURVE table (stage elevation / discharge pairs) for the control nodes are specified in the MAXSTAGE method.

4.2 Functional Solution Description

The first enhancement to be implemented is to move specification of the rating curve/table information from the RES-J MAXSTAGE method to the controlling nodes. Storing the rating curve/table information at the nodes will provide more physically appropriate definition of data because the information is specific to the nodes. Additionally, the capability to access a rating curve from the operational data will be incorporated (with the access occurring through specification of the rating curve ID at the nodes). This will assure any forecast will use the most up-todate database data in an operational system.

The second enhancement will be to allow the MAXSTAGE method to utilize a MAXIMUMDISCHARGE parameter (in addition to the current MAXIMUMSTAGE parameter). This will allow discharge-based operating rules to be implemented, rather than stage-based rules where stage/discharge relationships may change over time.

4.3 Inputs

The following section explains the parametric information required for the enhanced RES-J MAXSTAGE method. It specifies the relevant section of the NWSRFS User Manual Documentation that will change to reflect the enhanced RES-J MAXSTAGE capabilities. Changes to the existing document are denoted by highlights.

V3.3-RES-J-MAXSTAGE JOINT RESERVOIR REGULATION OPERATION METHOD MAXSTAGE

Method MAXSTAGE iteratively solves for the release of a reservoir until the downstream stage (corresponding to the routed release plus any additional inflows downstream of the reservoir) at a control point (node) is within a specified tolerance of a specified maximum value.

In essence, MAXSTAGE suspends simulation of the RES-J system, copies the affected portion of the system and runs the copy from the current time step forward sufficient time for the prescribed release to reach the control point. Simulation of the copy RES-J system is according to the same rules and Methods as for the original system. The prescribed release is adjusted and simulation repeats until an acceptable release is determined or the maximum number of iterations is reached.

<u>Input</u>

<u>Keyword</u>	<u>Definition</u>
MAXSTAGE	Indicates the start of parametric information for Method MAXSTAGE. 'MAXSTAGE' must be followed by: • a reservoir-type component identifier as defined in the TOPOLOGY section • an identifier for this Method
[TABLE]	Indicates the start of the stage/discharge rating curve table. TABLE must be followed by the ‘RATING CURVE’ keyword. Beginning

on the next line after ‘RATING CURVE’ create a table of stage elevation / discharge pairs describing the channel hydraulic conditions at the node. A rating curve TABLE cannot be specified if the DSCONTROL node has been parameterized with a rating curve, either using the RATINGCURVEID or TABLE keyword. Note: this section’s inputs are retained only for backward compatibility

[ENDTABLE]

Indicates the end of the rating curve table.

MAXIMUMSTAGE or
MAXIMUMDISCHARGE

Maximum stage or maximum discharge allowed at the downstream control point. Either MAXIMUMSTAGE or MAXIMUMDISCHARGE must be specified, but they cannot both be specified. Units are as described in the previous Parameter Definition section.

MINRELEASE

Minimum allowed reservoir release. This will likely be equal to MINRELEASE as defined in the reservoir parameter definition section, but may be different.

[CRITERION]

Convergence criterion. The value is used as an absolute allowable difference in simulated stage and MAXIMUMSTAGE where the value compared to CRITERION is calculated according to the equation:

D = abs((MAXIMUMSTAGE-SIM_STAGE)/MAXIMUMSTAGE)*100

DSCONTROL

Indicates the downstream control point. 'DSCONTROL' must be followed by:

- a node-type component identifier as defined in the TOPOLOGY section

[MAXITERATIONS]

Maximum number of iterations. If not user-defined, MAXITERATIONS defaults to 20.

ENDMAXSTAGE

Indicates the end of MAXSTAGE parametric information.

4.4 Error Messages

Error messages will be generated for the following error conditions:

1. Both MAXIMUMSTAGE and MAXIMUMDISCHARGE are specified (CONOPS 4.2.5)

** MAXSTAGE INPUT ERROR**

Both MAXIMUMSTAGE and MAXIMUMDISCHARGE are specified.

Action: Check MAXSTAGE method definition. Only one of MAXIMUMSTAGE or MAXIMUMDISCHARGE may be defined.

2. Neither MAXIMUMSTAGE or MAXIMUMDISCHARGE are specified

** MAXSTAGE INPUT ERROR**

Neither MAXIMUMSTAGE or MAXIMUMDISCHARGE are specified.

Action: Check MAXSTAGE method definition. One of MAXIMUMSTAGE or MAXIMUMDISCHARGE must be defined.

3. The RES-J MAXSTAGE method specifies a rating table and a rating table/curve id is also specified at the node (CONOPS 4.2.4)

** MAXSTAGE INPUT ERROR**

MAXSTAGE defines a rating table, and a rating table or curve ID are specified at the node (NODE_ID) as well.

Action: Check MAXSTAGE method parameters and NODE_ID node definition.

4. Stage constraint used but no rating curve ID or table found at the node

** MAXSTAGE INPUT ERROR**

Stage constraint used but no rating curve ID or table found at Node NODE_ID.

Action: Check MAXSTAGE method parameters and NODE_ID node definition.

4.5 Source Code

MaxStage_construct.cxx, MaxStage_solveMethod.cxx, MaxStage_initialize.cxx, MaxStage_Constructors.cxx, MaxStage.h, intfgq.f, intfrc.f, fstgq.f

4.6 Detailed Design

As much as possible, the implementation for the MAXSTAGE enhancements will make use of existing code.

When a rating TABLE is specified as input, an existing MaxStage routine that does a simple lookup and interpolation will be used. In both of these cases, the ultimate goal is to calculate a maximum flow that can be used in the existing MaxStage algorithm. After the maximum flow is calculated, no changes will need to be made to the MaxStage algorithm (except to allow for specification of a MaximumDischarge instead of a MaximumFlow).

Note that if a rating curve id is defined at the node, the relevant parameters (NCROS, FRLOOP, EMPTY, etc) from the FRATNG Common Block will be used by STAGE-Q during the conversion.

The following source code files are involved in the implementation of the MAXSTAGE enhancements.

4.6.1 MaxStage_Construct.cxx

This routine reads from the input deck and initializes the MaxStage object.

Changes required:

1) Allow the TABLE input parameter to be optional instead of required (the rating table used by the MAXSTAGE method can be specified at the node explicitly, or using a rating curve id). For backward compatibility, specifying the rating table within the method will continue to be supported. If a TABLE is not defined for the method, but a TABLE is defined at the node, populate the _stage_flow_tbl with the data from the node.
2) If TABLE is defined for the method, and a rating table or curve ID is defined at the downstream control point (node), print an error and exit
3) Allow for the specification of a MAXIMUMDISCHARGE parameter for the MAXSTAGE method.
4) Check for (and print an error message) if either of the following conditions occurs
    a. Neither MAXIMUMSTAGE nor MAXIMUMDISCHARGE is defined
    b. Both MAXIMUMSTAGE and MAXIMUMDISCHARGE are defined

4.6.2 MaxStage.h

Header file.

Changes required:

- Add _max_discharge data member to the MaxStage class.
- Add _max_flow data member to the MaxStage class.

4.6.3 MaxStage_initialize.cxx

This routine initializes a new MaxStage object (called by the constructor).

Changes required:

- Initialize _stage_flow_table to NULL
- Initialize _max_discharge to MISSING

4.6.4 MaxStage_Constructors.cxx

Constructors for a MaxStage object.

Changes required:

- When creating a new MaxStage method via copy, set this object's _max_stage and _max_discharge members to the appropriate values (of the source object)

4.6.5 MaxStage_solveMethod.cxx

This is the primary routine used in determining the MaxStage solution.

This routine will be used essentially intact with the most significant changes being: do not do the lookup to convert a _max_stage to a _max_flow if the MAXIMUMDISCHARGE parameter is defined (instead of the MAXIMUMSTAGE parameter); and, call into the STAGE-Q framework routine to do the stage/discharge conversion when a rating curve ID has been specified at the downstream node. When a rating TABLE is defined at the node (or for the method), the existing MaxStage lookup algorithm will be used. In other words,

```
// determine the maxflow to test for this iteration
If _max_discharge != MISSING then
   // MAXIMUMDISCHARGE was defined for the method
   maxflow=_max_discharge
else if _stage_flow_table != NULL
   // the rating curve TABLE was defined in the method or
   // at the node; use the existing algorithm and methodology
   // to determine maxflow
   maxflow=_stage_flow_table.lookup(_max_stage)
else
   // not a discharge based evaluation; a rating TABLE has
   // not been defined; must be doing a lookup based on
   // a rating curve id defined at the node
   maxflow=(call into the STAGE-Q FSTGQ routine to lookup a flow
            as a function of _max_stage and the rating curve ID)
endif

// continue with the method, using maxflow as the release to
// be tested
```

4.6.6 fstgq.f, intfgq.f, intfrc.f

In order to call into the STAGE-Q FSTGQ routine (which is in FORTRAN) from MaxStage_solveMethod.cxx (C++) an appropriate interface will be developed. As part of the interface definition, the FSTGQ input parameter list needs to be populated within the C++ environment. Below is a table listing the parameters and their definition/source:

Parameter Name	**Definition**	**Source notes for C++ implementation**
RCID	Rating Curve ID	Defined at the node; _dcp

		(downstream control pt)
ICONV	Conversion Indicator (0=Stage->Discharge; 1=Discharge->Stage)	Always 0
ITSPOS	Initial time series position to convert	Only converting one timestep; based on current Julian Day; _solveMethod cur_date var.
NVALS	Number of values to convert	Always 1
TSDELT	Time step	MaxStage.getTimeInterval();
QDATA/HDATA	Discharge/Stage arrays	Used by STAGE-Q for timeseris based input/output (independent/dependent var); 1-element arrays used here
LOCPTR	Pointer array for loop rating curve program	NULL*
T1	Timing array for loop rating curve program	NULL*
CURVLO/HI	Lowest/highest rating curve stage	-999 (set within FSTGQ)
XSECLO/UP	Lowest/highers cross-section eleveation	-999 (set within FSTGQ)
METHOD	Conversion Method (0=simple interpolation/extrapolation: 1=dynamic loop)	0 **
FLSTAG	Flood stage (M) wrt gage zero	-999 (N/A)
NEEDEX	Output param indicating extension required	-999; return values of 6, 7,or 8 indicate looping – will generate an error message
CARRYO	Carryover array	(0,0,0,0) C/O not supported
JULDAY	Julian day of initial value to convert	_solveMethod cur_date
INITHR	Hour of initial value to convert	_solveMethod cur_date
IRCHNG	FRATNG common block xfer indicator (output)	-999
IERROR	Error flag (output)	N/A
IPRWRN	Warning msg flag (output)	N/A

*Loop rating not supported in this enhancement, but could be subsequently enabled

**The METHOD variable may be changed to 1 within the FSTGQ routine in response to the value of the FRLOOP variable in the FRATNG common block. If this occurs, it will be flagged and an error will be generated – rating curves specifying a dynamic loop are not supported as part of this project.

intfrc.f and intfgq.f are C++ to Fortran wrappers developed to ease the transition between C++ and Fortran.

The new fstgq.f is used when the mcp3 program is run and an attempt is made to access a rating curve by its id (an unsupported operation). The new fstgq.f will report that it is not active and return.