

ResSIM lite description and documentation. The model is based very closely on the ResSIM model developed for the Willamette by the USACE. In fact, the purpose of this model is to represent, to the greatest extent possible, ResSIM within the Envision framework. Project Contact: Desiree Tullos, OSU (desiree.tullos@oregonstate.edu).

1. Information about the reservoirs is read in via an xml file.

Function: FlowProcess:Load Xml

1.1 Load Reservoir attributes

- The xml file contains the following information about each reservoir. This information is uploaded and stored in data objects associated with the Reservoir class

Variable name	Type	Description
Name	Cstring	Reservoir name
ID	Int	Reservoir ID 1-100 USACE 100-200 EWEB 200-300 PGE 300-400 USBR
Dir	Cstring	Path of reservoirs main directory
InitVolume	Float	Initial volume, set to lower value of flood control pool, m3
Area_vol_curve	Cstring	Filename of text file with area, volume and pool elevation relationships
Av_dir	Cstring	Directory containing area volume curves
Rule_curve	Cstring	Filename of text file with rule curve information
Rc_dir	Cstring	Directory containing rule curves
Composite_rc	Cstring	Filename of text file with release capacity information
Re_dir	Cstring	Directory containing release capacity curves
Rule_priorities	Cstring	Filename of text file with rule priority information, by zone (e.g. flood control, conservation, buffer)
Rp_dir	Cstring	Directory containing rule priority file
Cp_dir	Cstring	Directory containing text files with control point constraints (accessed by multiple reservoirs)
minOutflow	Float	Minimum outflow (cms)
maxVolume	Float	Maximum reservoir volume (m3)
Td_elev	Float	Top of dam elevation (m)
Inactive_elev	Float	Inactive elevation (m)
Fc1_elev	Float	Flood control elev (m) – only if more than 1 flood control zone exists
Fc2_elev	Float	Flood control elev (m) – only if more than 2 flood control zones

		exist
maxPowerFlow	Float	Physical maximum flow that can be passed through the powerhouse
RO_rc	Cstring	Filename of text file containing capacity information for regulating outlets (typically dependent on pool elevation).
Spillway_rc	Cstring	Filename of text file containing capacity information for spillway outlets (typically dependent on pool elevation).
Tailwater_elev	Float	Elevation of tailwater (m). Used in hydropower calculations.
Turbine efficiency	Float	Reported turbine efficiency (values taken directly from ResSIM). Used in hydropower calculations.
Buffer_zone	Cstring	Filename of text file containing the buffer zone curve values
Ressim_output_f	Cstring	Filename of file containing flow output from ResSIM model (used for model comparison purposes)
Ressim_output_r	CString	Filename of file containing rule output from ResSIM model (used for model comparison purposes)

Table 1 - Reservoir attributes contained in the .xml file.

1.2 Load Control Point attributes

- the xml file contains the following information about each control point. This information is uploaded and stored in data objects associated with the Control Point class

Variable name	Type	Description
name	Cstring	Name of control point (names used directly from ResSIM)
Id	INT	Unique numerical ID
Dir	Cstring	Directory where the text file describing constraints at the control point is located
Control_point	Cstring	Name of the text file describing constraints
Location	Int	COMID of the reach within which the control point is located
Reservoirs	Cstring	A list of all the reservoir that this particular control point influences

Table 2 - Control Point attributes contained within the .xml file.

2. Reservoirs are initialized

Function(s): FlowModel:Init->InitReservoirs

2.1 Identify Reservoir locations

Iterate through stream layer polylines and find those with ResIDs. These indicate locations of the reservoirs. Only one reach with an associated ResID is needed. After one reach in each of the reservoirs is located, traverse downstream until the next downstream node is not a reservoir (or a different reservoir). This is the farthest DS reach in the reservoir. Store this location in pRes->m_id. (pRes is throughout the model a pointer to a reservoir).

2.2 Load Inflow values from HEC ResSIM

This block of code is only used if the model will be run using the reservoir inflows from HEC ResSIM, as opposed to those generated by another hydrology model within Envision. Discharge values at reservoir inlets are simply replaced with values from an input time series

2.3 Load Tables describing reservoir operations rules

The following tables are loaded into data objects for recall in the model. 1) Area Capacity Curve, 2)Reservoir rule curve, 3) Release Capacity curve, 4) Rule Priority Table. The minimum outflow value for each reservoir is also initialized here.

2.4 Process the rule priority table

2.4.1 Data Storage Structure for Reservoir Rules

The rule priority table contains a list of constraints, from lowest priority on top to highest priority on the bottom, for each operation zone within the reservoir. The zones are organized as follows:

Zone 0	Top of Dam
Zone 1	Flood Control
Zone 2	Conservation
Zone 3	Buffer
Zone 4	Alternative Flood Control 1
Zone 5	Alternative Flood Control 2

Table 3: Reservoir Zones in ResSIM Lite

Example rule priority file :

0	1	2	3	4
Maxl_FloodIncrsRamp_Cougar.csv	Max_Flow_Winter_and_Conservation.csv	cp_Min_Flow_at_Albany_23762845.csv	Min_RO_flow.csv	Min_Low_RO_flows_turbine_low.csv
Max_Flow_Winter_and_Conservation.csv	cp_Max_Vida_Flood_23772903.csv	cp_Min_flow_at_Salem_23791083.csv	MaxD_Revised_Daily_BiOP_MaxD.csv	Pow_Min_Low_Turbine_Flows_at_Low_Res_Elev.csv
MaxD_FloodDcrsRamp_Cougar.csv	Min_ConservFlow_Cougar.csv	cp_Max_Vida_Flood_23772903.csv	Min_ConservFlow_Cougar.csv	
RO_Min_flow.csv	Maxl_High_Flow_Ramp_Up.csv	MaxD_Revised_Daily_BiOP_MaxD.csv	Pow_Min_CGR_power_release.csv	
Pow_Min_CGR_power_release.csv	MaxD_FloodDcrsRamp_Cougar.csv	RO_Min_flow.csv	Pow_Max_CGR_updated_power_release.csv	
Pow_Max_CGR_updated_power_release.csv	RO_Min_flow.csv	Maxl_IncrsRamping_Cougar.csv		
	Pow_Min_CGR_power_release.csv	Max_Flow_Winter_and_Conservation.csv		

Table 4: Structure of the rule priority table (in this case, Cougar Reservoir)

The zone number is in the first row, and the names of the text files corresponding to each constraint are located in a list below. Each column in the table above is stored as an array of objects called ResConstraints within an object called a ZoneInfo. A ZoneInfoArray for each reservoir stores the array of ZoneInfo objects (5 in the example above). Each ResConstraint is a text file with allowable reservoir release values given 1 or 2 variables. There are 8 types of constraints described in the table below:

Rule type	File header	pConstraint->m_type
Minimum	Min_	1
Maximum	Max_	2
Increasing Rate	MaxI_	3
Decreasing Rate	MaxD_	4
Control Point	cp_	5
Power Plant rule (gate specific)	Pow_	6
Regulating Outlet rule (gate specific)	RO_	7
Spillway Rule (gate specific)	Spill_	8

Table 5: Rule types and file headers that denote the rule type

For example, a minimum flow ResConstraint (MaxD_Revised_Daily_BiOP_MaxD.csv) that varies by reservoir outflow at the previous timestep:

Outflow_lagged_24h	Min_release_cms
11	11
23	11
51	25
96	51
159	96
232	159
320	232
422	320

Table 6: Example of reservoir constraint file

2.4.2 Process Rule Priority Function

Function: Reservoir->ProcessRulePriorityTable

- First, the function retrieves the dimensions of the rule priority table, creates a ZoneInfo for each zone and stores them in a ZoneInfoArray.
- Next the header is parsed to determine whether to look in the rules folder for this particular reservoir or the control points folder (which contains constraints shared by more than one reservoir). The header will be "cp_" for control point rules.

- Iterating through all the rules, the text file corresponding to each rule is loaded into a data object for retrieval by the Envision model
- The header is parsed again, this time to determine the rule type. All the rule filenames have a header which signifies the rule type. An integer is assigned to each constraint to denote its type. The file headers and corresponding integers are detailed in Table 3.
- Each constraint is added to a ResConstraintArray (which is contained with a ZoneInfo object).

2.4.3 File Directories

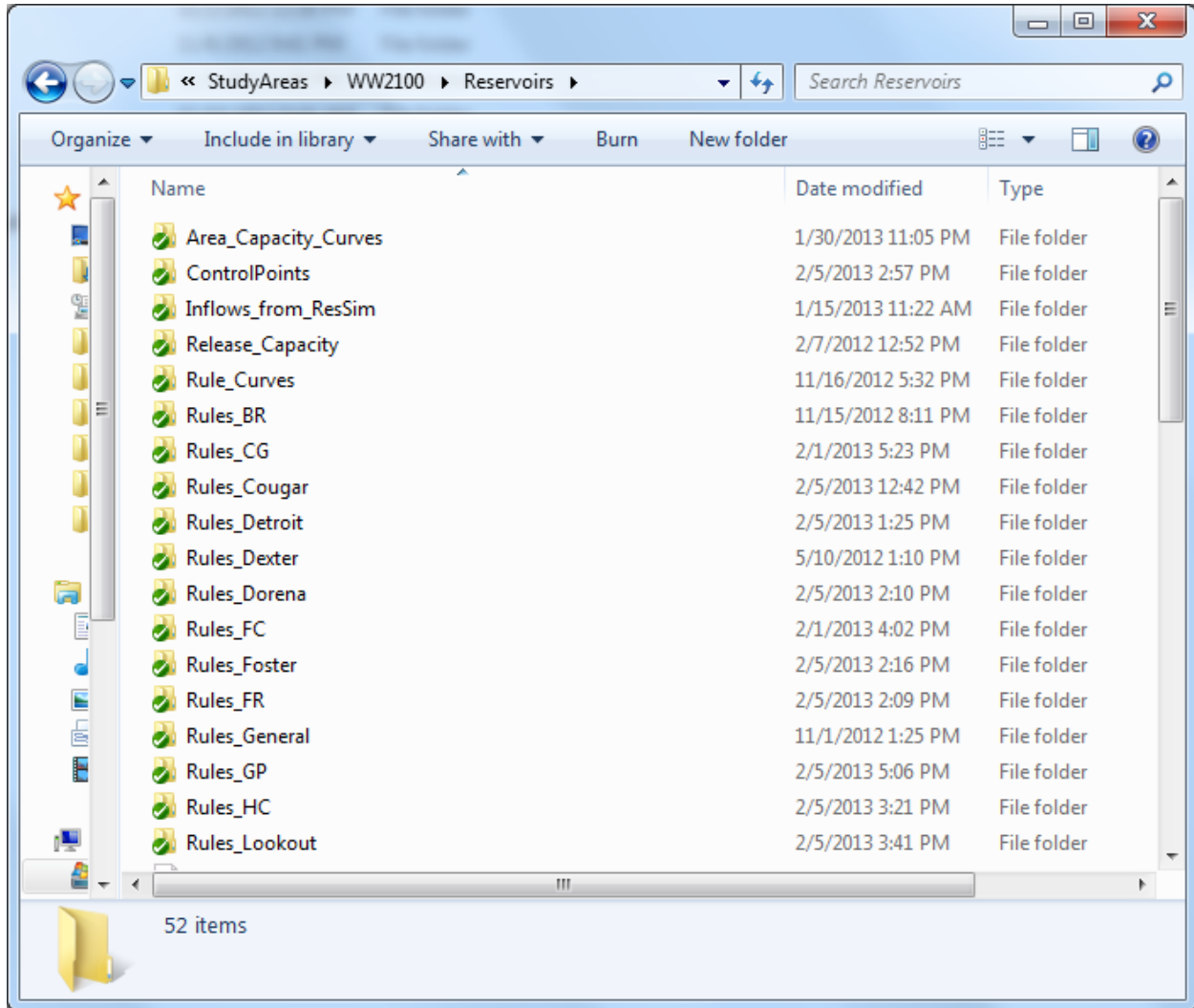


Figure 1: Reservoirs folder located in the Envision\StudyAreas\WW2100 directory. Separate directories exist for Rule Curves, Area Capacity Curves, Control Points and rule-sets for each Reservoir. (Full names are not spelled out for each reservoir due to string size limitations.

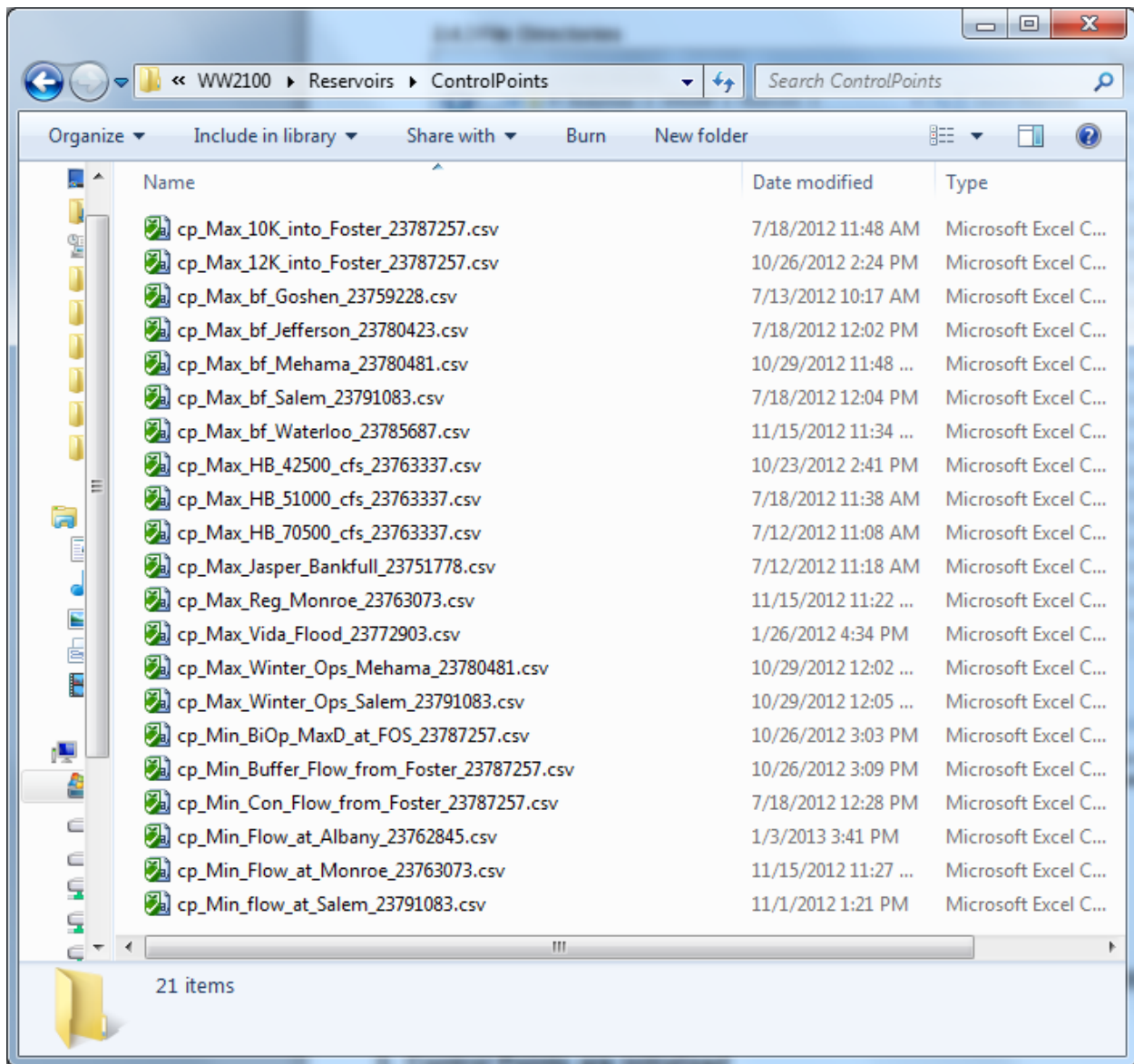


Figure 2: Control Points folder located in the Reservoirs directory. Note how multiple types of control points at the same location (e.g. min and max) have separate files. Each file has a number at the end of it, which corresponds to specific reach location (COMID) in the streams coverage. This number is read by envision in order to assign a spatial location to each control point.

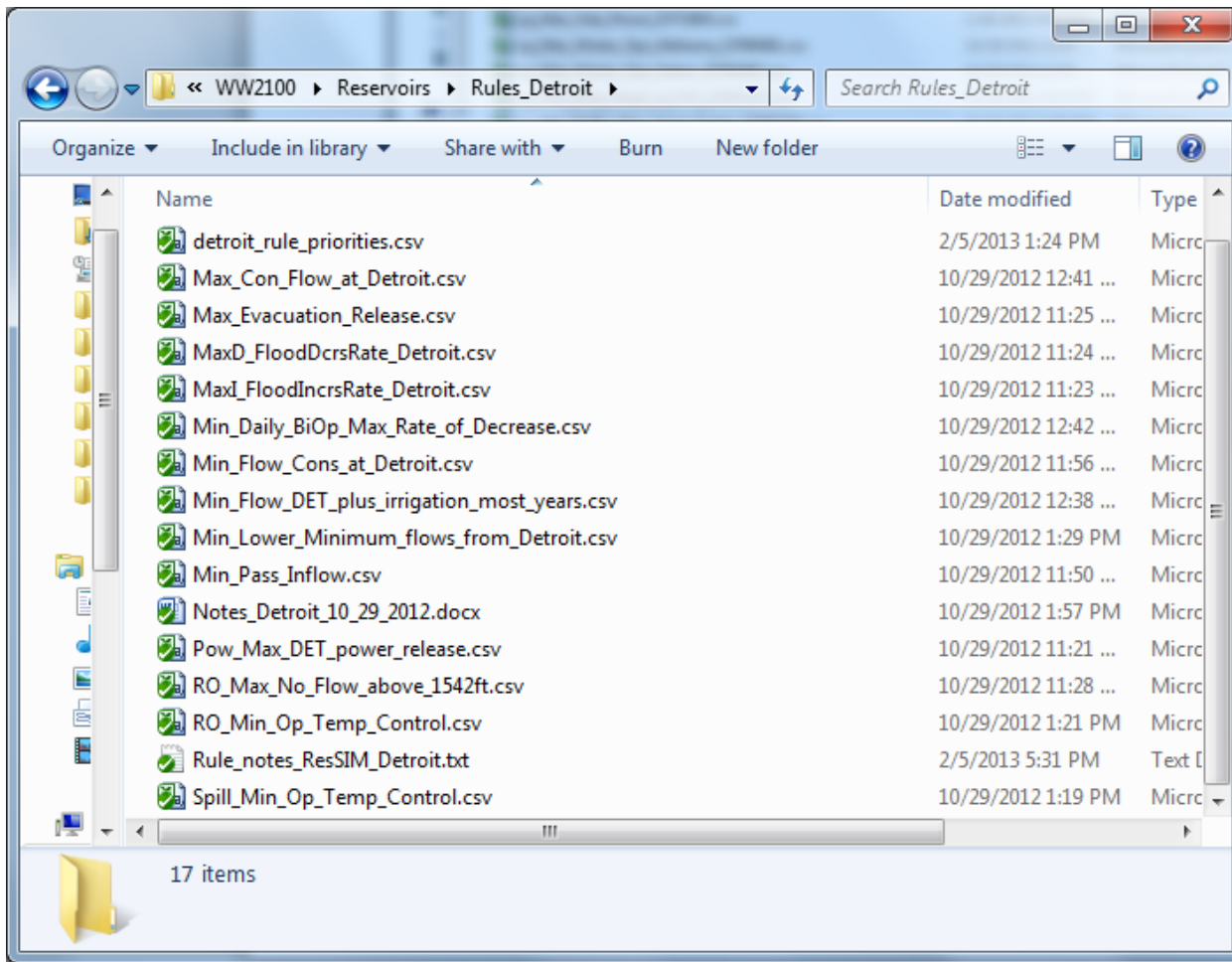


Figure 3: An example of the contents of a rules folder for a specific Reservoir (in this case Detroit). The file entitled “Detroit_rule_priorities.csv” is structured similarly to the example in Table 2. Each rule contains a tag at the beginning of the filename to categorize it (e.g. min, max, pow, RO, etc..) as described above. Names are as consistent as possible with rule names used in ResSIM, although some had to be shortened due to string length limitations.

2.5 – Initialize data collection

Function: Reservoir->InitDataCollection

Initializes data objects that will be used to store output data for display in Envision. Each reservoir modeled has one data object that stores 8 variables (time, pool elevation, rule curve elevation, inflow, outflow, powerhouse flow, regulating outlet flow and spillway flow). If implemented, there are also 2 files initialized to compare the results of ResSIM with the results of Envision. One (ResSimFlowCompare) stores pool elevation, inflows and outflows from both ResSIM and Envision, while the other (ResSimRuleCompare) stores the active rule (constraint that controls outflow) at each timestep from both model. (Note, ResSIM has been pre-run and the output data stored in a text file, which is read in to Envision to make these comparisons).

2.6 – Import Inflows from ResSIM

This step only occurs for model comparison with ResSIM outputs. When comparisons are being made, the inflows at each reservoir are replaced with inflows from the ResSIM model to facilitate comparison between the two models. This violates the network mass balances, and should only be used for testing.

3. Control Points are initialized

Function: FlowModel->InitReservoirControlPoints

First, the size of the array of control points is retrieved. Then the function loops through all the control points and does the following for each individual control point:

- Creates a new Data Object called m_pResAllocation
- Reads the string of Reservoirs influenced that was previously loaded from the xml file
- Parses the string and creates an array of pointers to reservoirs influenced (pControl->m_influencedReservoirsArray) by this particular control point
- Loops through all stream reaches and associates a reach (pControl->m_pReach) to each control point via matching comIDs. (Control point COMIDs read from the xml input file).
- A table of constraints (based on date, pool elevation, etc..) is loaded for each control point.

Note: Because each control point is set up to have only 1 table of constraints, different types of constraints at the same location show up as separate control points, with separate constraint tables (e.g. Min flow at Salem and Max Bankfull Flow at Salem).

4. Calculate Reservoir Fluxes

Function: FlowModel->Run->SetGlobalReservoirFluxes->SetGlobalReservoirFluxesResSIMLite

4.1 UpdateReservoirControlPoints

Function: FlowModel->UpdateReservoirControlPoints

This function iterates through each control point in the model and updates the amount of flow allocated from reservoirs that are influenced by the particular control point. Many reservoirs can be influenced by a single control point (e.g Min Flow at Salem) and many control points can be influenced by a single reservoir. At this point (June 4, 2013) storage ratios within the basin are not weighted, and therefore each influenced reservoir contributes equally to keep control point flows within their targets.

Case 1: Flow above the maximum. If the flow at a control point is above the maximum allowed (bankfull), then negative amounts are allocated from each reservoir influenced, and outflow is reduced at each of these reservoirs by the specified amount to help bring the control point back within its limits.

Case 2: Flow below the minimum. If the flow at a control point is below the minimum allowed, then positive amounts are allocated to each reservoir influenced, and outflow is increased at each of these reservoirs by the specified amount to help bring the control point back within its limits.

Case 3: Flow is above the minimum and below the maximum. Zero flow is allocated from the influenced reservoirs.

ResSIM comparison: In the case of model runs to compare with ResSIM results, the discharge at each control point is augmented by a "local flow" value. (These values are taken directly from the ResSIM inputs). This should be used

in conjunction with a scenario of minimal simulated inflows from Envision. (Local flows are values of “cumulative local flow” at each control point in ResSIM, and therefore represent the influence of all uncontrolled upstream inputs at that particular control point).

4.2 UpdateReservoirWaterYear

Function: FlowModel->UpdateReservoirWaterYear

On May 20th (day of year 140), the volume of water stored in all of the reservoirs is summed. A water year type is then assigned based on the following table (from the 2008 Biological Opinion).

Characteristics of Water Year Types	Abundant	Adequate	Insufficient	Deficit
Mid May Storage (MAF)	≥ 1.48	1.20 to 1.47	0.90 to 1.19	< 0.90

Table 7 - Water Year Types, based on total storage volume in all USACE reservoirs on May 20th of each year.

4.3 GetResOutflow

Function: Reservoir->GetResOutflow

This is the main function containing the application of the rules. The function takes a pointer to the reservoir and the day of year and returns an outflow volume.

Narrative Description of the Outflow Decision Making Process:

4.3.1 – Update elevation and rule curve information and set desired release

First, the current pool elevation is obtained (Reservoir->GetPoolElevationFromVolume). Maximum gate outflows are updated based on the current pool elevation (Reservoir->UpdateMaxGateOutflows).

If the reservoir is a re-regulating reservoir, inflow is set equal to outflow and the rule process is skipped.

Update the following values: L Point to rule curve elevation (target pool elevation), buffer zone elevation, target pool volume, and current volume.

The desired release is simply the amount of water needed to be released to take the pool elevation back to the target pool elevation in one timestep. (If we are below the target pool elevation, then the desired release is set to 0).

4.3.2 – Define the operations zone

The operations zone defines which constraints (rules) are to be used to determine outflow. The operation zone is defined by the current pool elevation with respect to the rule curve (conservation – below in green). The zone above the rule curve is the *Flood Control Zone*, while below the curve is the *Conservation Zone*. There exists an alternative flood control zone in this case, when the pool elevation is above the rule curve but below the secondary flood control line (in light blue). If the pool elevation is above the elevation demarcating the primary flood control zone (horizontal dark blue line), the zone is labeled *Top Of Dam Zone*. If the pool elevation falls below the red line, the zone is labeled the *Buffer Zone*.

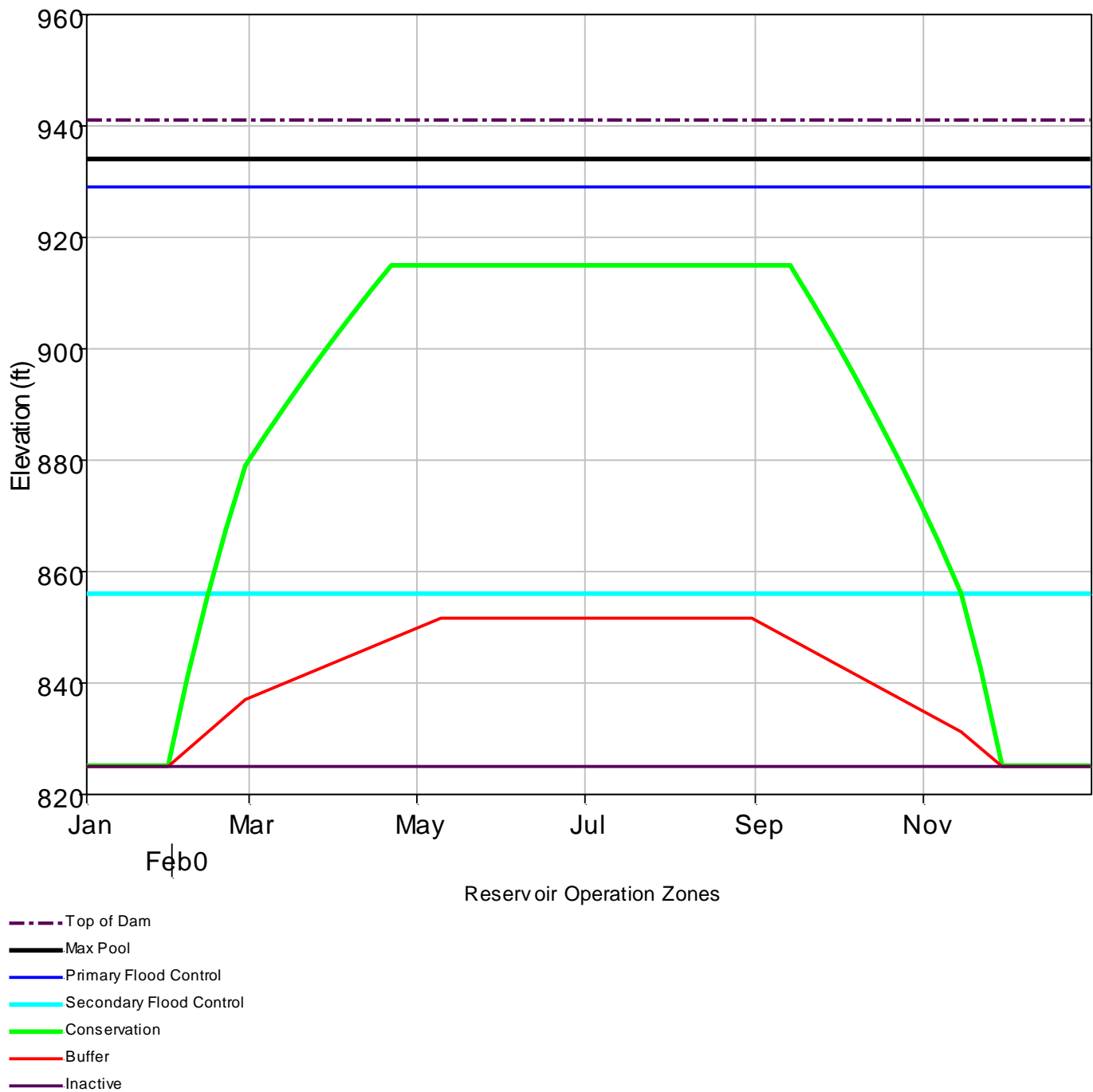


Figure 4 - Reservoir operating zone (in this case shown for Lookout Point). The conservation curve (green) is the primary rule curve.

Once the operating zone is determined, constraints are applied in order (determined by their place in the column of the rule priority table (Figure 2).

4.3.3 - Apply constraints, prioritized within each zone

The text in the first row and first column of each rule text file contains the value upon which the rule depends (all values stored within a ResConstraint data object). The values currently accepted are listed in the table below. Note that the text must match exactly for the rule to function properly.

String notation	Lookup value
-----------------	--------------

"Date"	Day of year (1 to 365)
"release_cms"	Outflow at last timestep
"pool_elev_m"	Current pool elevation
"inflow_cms"	Current inflow value
"Outflow_lagged_24h"	Outflow at last timestep (this notation is taken from the ResSIM model, but with a daily timestep, this is functionally equivalent to a "release_cms" based rule)
"date_pool_elev_m"	Lookup is based on 2 values (date and pool elevation)
"date_water_year_type"	Lookup is based on 2 values (date and water year type)
"date_release_cms"	Lookup is based on 2 values (date and outflow at last timestep)

Table 8 - Allowable headings for the first column in rule text files.

Constraint values are obtained from the lookup tables for each individual rule. If the constraint value applies (e.g. the outflow is greater than a maximum, less than a minimum, etc.), then the actual release is adjusted to conform to the constraint. The constraints are applied in order of priority, from lowest to highest. The logic describing release decisions is explained in detail in section 11.4.1 of the HEC-ResSIM User's Manual.

There are 8 types of constraints current defined:

```
enum RCTYPE { // ResConstraint Types
    RCT_UNDEFINED      = 0,
    RCT_MAX            = 1,
    RCT_MIN            = 2,
    RCT_INCREASINGRATE = 3,
    RCT_DECREASINGRATE = 4,
    RCT_CONTROLPOINT   = 5,
    RCT_POWERPLANT     = 6,
    RCT_REGULATINGOUTLET = 7,
    RCT_SPILLWAY       = 8
};\
```

Max and Min constraints set upper and lower boundaries on outflow. Increasing rate and decreasing rate rules govern the allowed rate of change of flow releases (these constraints would also be called *ramping rates* in discussions of reservoir operations. The control point constraints adjust outflow based on current conditions at the control point. The final three types (POWERPLANT, REGULATINGOUTLET, AND SPILLWAY) do not govern total outflow allowed from the reservoir but are operational constraints on the type of outlet in question.

Each time a constraint changes the outflow value, the name of the constraint is stored as the "active rule". The final constraint applied (and thus, the constraint controlling the outflow value) is stored for QA/QC and comparisons with ResSIM.

After application of all the rules, outflow is checked to see if it is below the allowed minimum flow (which is enumerated in the .xml file (see Table 1). If the pool elevation falls below the inactive level (also specified for each reservoir – see Table 1), then release of more water is physically impossible and the outflow is set to zero.

4.3.4 – Assign Reservoir Outlet Flows

Once the outflow for a particular reservoir has been determined, the outflow is split between the outlet gates, which can be either power plant outlets, regulating outlets, or spillway gates. All the reservoirs in the model are prioritized in the same way, with outflows being routed first to the powerhouse, next to the regulating outlets, and finally to the spillways. This sequential release allocation strategy is represented in exactly the same way in the ResSIM model. Outlet gates are subject to physical and operational constraints as defined either in the xml file (physical), or within the prioritized list of constraints (operational).

5. Collect Output Data

Function: FlowModel->Collect Data

Output data is collected for

- a) Display within the Envision Model post-run results table,
- b) For comparison of the output with ResSIM output.

The following values are collected daily at each reservoir modeled for display in Envision:

Pool Elevation, Rule Curve Elevation, Inflow, Outflow, Powerhouse Flow, Regulating Outlet Flow and Spillway Flow.

For comparison with ResSIM, pool elevation, inflow and outflow are stored, alongside these values from ResSIM (preprocessed and read in from a text file). Additionally, the name of the “active rule” described above in section 4.3.3 is stored alongside the active rule from the ResSIM model.