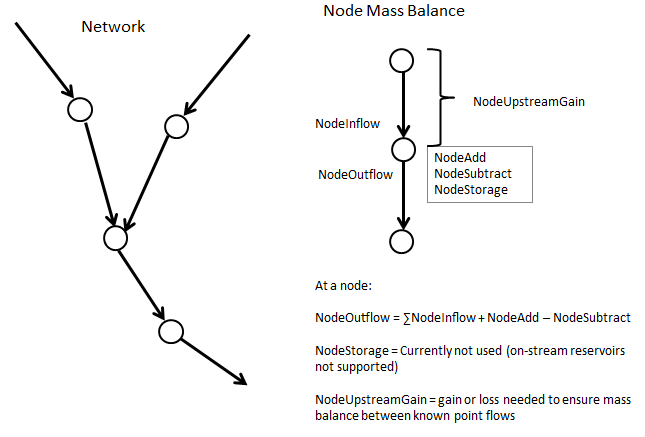
Command Reference: AnalyzeNetworkPointFlow()

Analyze a node/link network to calculate “point flow” for nodes

Version 10.21.00, 2013-07-02

The AnalyzeNetworkPointFlow() command takes as input information to define a “flow network”, associates input time series with each node in the network, and computes output time series at each node. Although the network is intended to represent a physical network such as a stream system, it also can represent other flow networks such as transportation or other mass/energy conservation systems. The following figure illustrates the flow network connectivity and mass balance that is performed at each node. Currently “on-channel” reservoirs with storage are not supported and gain/loss can only be computed in non-branching networks – these features will be added in the future. Consequently the command is suitable for a main stem river with no on-channel reservoirs.



AnalyzeNetworkPointFLow

AnalyzeNetworkPointFlow() Network and Node Mass Balance

The AnalyzeNetworkPointFlow() command uses input time series associated with the nodes to compute mass balance at each node, and creates output time series with the following data types:

NodeInflow – sum of outflows from upstream nodes

NodeAdd – time series added at the node

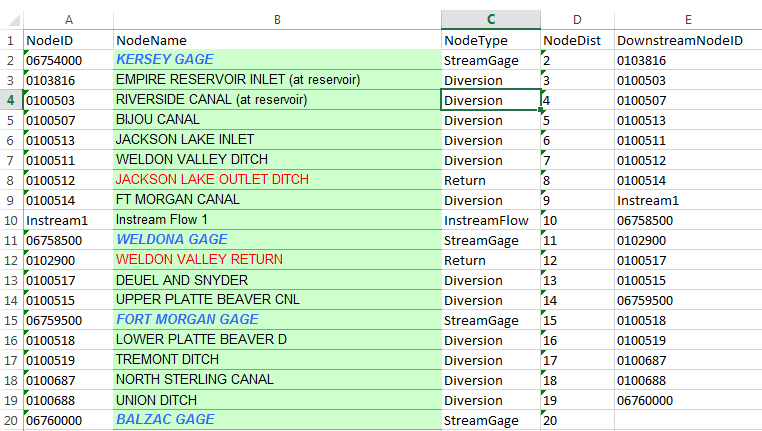
NodeSubtract – time series subtracted at the node

NodeStorage – storage at the node after additions and subtractions (currently always zero, will enhance in the future to handle on-channel reservoirs)

NodeUpstreamGain – gain or loss (negative) between upstream node(s) and the current node (zero unless gain/loss is computed)

NodeOutflow – outflow from the node

The network is defined as a table containing a list of node identifiers with associated properties, as illustrated in the following figure.



AnalyzeNetworkPointFLow

AnalyzeNetworkPointFlow() Network Input Table

In this example the network is defined in an Excel file, the ReadTableFromExcel() command is used to read the table, and the table is used as input to the AnalyzeNetworkPointFlow() command. This command differs from the functionality of other network analysis tools as follows:

* Daily administration tools, such as the State of Colorado’s Colorado Water Rights Administration Tool (CWRAT) perform a point flow analysis for a single day, which only requires knowing one day’s input values, whereas AnalyzeNetworkPointFlow()analyzes time series for a specified period.
* More sophisticated models, such as the State of Colorado’s StateMod water allocation model, perform allocation decisions within each timestep for the full period, whereas AnalyzeNetworkPointFlow()performs a sequence of basic time series manipulations that can be quickly configured.

It may be possible to utilize the network data from tools such as those mentioned above with the AnalyzeNetworkPointFlow() command.. The network definition table columns from the above figure are as follows:

| Column | Description |
| --- | --- |
| NodeID | The location ID for the network node, typically corresponding to the location ID in time series identifiers. |
| NodeName | The node name, useful because NodeID is generally terse and nondescriptive. |
| NodeType | The node type, needed to define node behavior (e.g., whether time series values get added, subtracted, reset at node). The node types are user-defined, although types often are defined by modeling conventions. The behavior of node types is defined by using command parameters (e.g., see NodeAddTypes). |
| NodeDist | The node distance along the flow path. Typically the distance is measured relative to the lowest point on the network. The distance is used to estimate gain/loss when GainMethod is specified as a command parameter. |
| DownstreamNodeID | The location ID for the downstream node, needed to define network connectivity. |

The following logic is used to analyze the network. Currently this logic is performed by navigating the network from most upstream to downstream and processing all timesteps for a node before moving to the next node. This may need to change to processing the analysis period and navigating the network for each time step. Handling dry-up points for gain/loss computations will impact the logic.

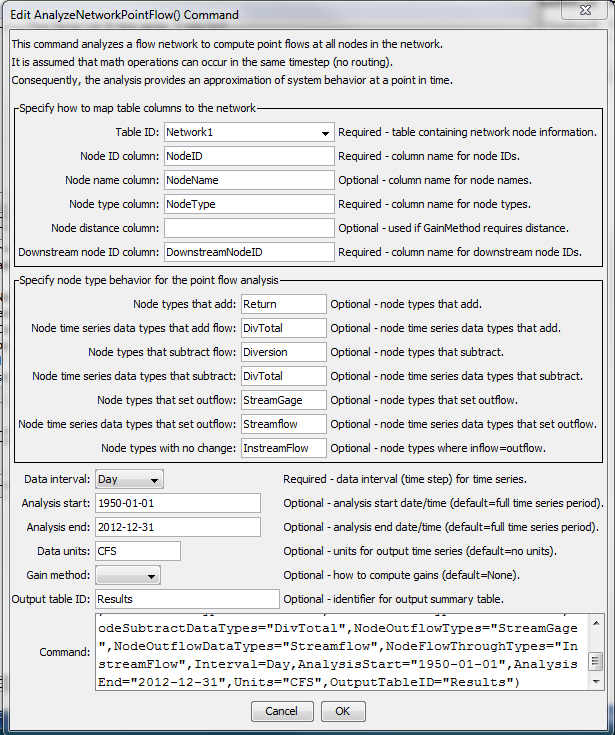
1. The network is navigated from top to bottom. When a confluence is found (a node with more than one upstream node), each confluence is processed from the top down to the confluence point. Of particular importance is the concept of a “stream reach”, which is the reach between known flow points, because mass balance is enforced at known flow points and gain/loss can be estimated between the known flow points.
   1. The data type for the node (see \*DataType command parameters) is used to retrieve the relevant time series for the node. The first time series that matches the location ID and data type is used as input for the node. The time series must have been read prior to the AnalyzeNetworkPointFlow() command. For example, use the CopyTable() command to copy a subset of the network table’s NodeID values and then use the ReadTimeSeriesList() command with the list of identifiers.
   2. The node is analyzed based on the node type:
      1. Node types that add (e.g., Return, Import):
         * NodeOutflow = NodeInflow + added time series
      2. Node types that subtract (e.g., Diversion):
         * NodeOutflow = NodeInflow - subtracted time series
      3. Node types that set outflow (e.g., StreamGage or simple Reservoir):
         * NodeOutflow = NodeInflow = outflow time series for node
      4. Node types that let flow through (e.g., InstreamFlow):
         * NodeOutflow = NodeInflow.
   3. If gain/loss is being estimated:
      1. When a node that has known outflow is encountered (e.g., StreamGage), gain/loss between this node and the nearest upstream node(s) is computed. **This has only been implemented for the case where all intervening nodes are in a non-branching reach**.
         * If the GainMethod=Distance, use the node distance data from the network table to prorate the gain/loss in the stream reach.
         * If the GainMethod=None, no adjustment to flows is made and the gain/loss upstream of the know flow node will result in a discontinuous jump because no gain/loss adjustment is made.
         * If the GainMethod=Weight, the gain/loss is prorated by the weights specified by the NodeWeightColumn parameter (or weight equally if the weights are not specified in the network table). The weight of the upstream stream gage is not used.
2. Analysis statistics are optionally written to an output table, which contains a row for each network node. Statistics include information such as the number of missing values in the input time series. This information can be used to evaluate the quality of the analysis. **This feature has not yet been implemented.**

Issues that need to be considered include:

1. Missing data in input result in missing data in calculated values. Use TSTools features to fill missing data in time series before using as input to the analysis. Because this may be a major effort, especially for a long analysis period, it may be appropriate to read time series from model data sets. It is envisioned that the output table will provide feedback on how much missing data there is and how it impacts the analysis.
2. TSTool’s graphing tool currently does not allow graphing lines as a step function in the case where no gain/loss is computed. Instead, the line connects the data points. An enhancement to the graphing tool is needed.
3. TSTool does not provide a way to graph a stream reach where the graph values are pulled from each time series for a point in time. Ideally a visualization tool would allow “scrolling” through dates and showing the river reach with flow on the Y axis and node distance on the X axis, although it would be tedious to have to scroll through the period.
4. There may be cases where a subtraction at the node takes all of the flow, essentially causing the node to be a known zero point flow. For example, in Colorado, a river call may result in a river drying up during the call. It is possible to estimate when this occurs, but the data quality may be low. The question is… should this command allow “zero flow” data to be specified, which would result in Diversion node values being treated as known point flow (NodeOutflow=0) and if so, how to specify the input data in a general way?

It is important to understand that such a point flow analysis represents a snapshot of the system at any point in time, but does not route flows through the network. Known flows at stream gages are used as fixed values from which other data are estimated. Gains and losses are representative of the network system, essentially interpolating over time and distance. This type of analysis introduces errors in cases where the lag time between nodes would result in significant differences if lagging were considered. In the physical system, changing an upstream flow would result in lagged impacts due to routing; however, the point flow analysis shows the impacts to downstream nodes in the same time step. A more sophisticated model with routing would be needed to represent actual conditions. However, the point flow analysis will be reasonably accurate if gains and losses are occurring because of fairly static phenomena (e.g., groundwater interactions that do not change rapidly within the network travel time).

The following dialog is used to edit the command and illustrates the syntax of the command:



AnalyzeNetworkPointFLow

AnalyzeNetworkPointFlow() Command Editor

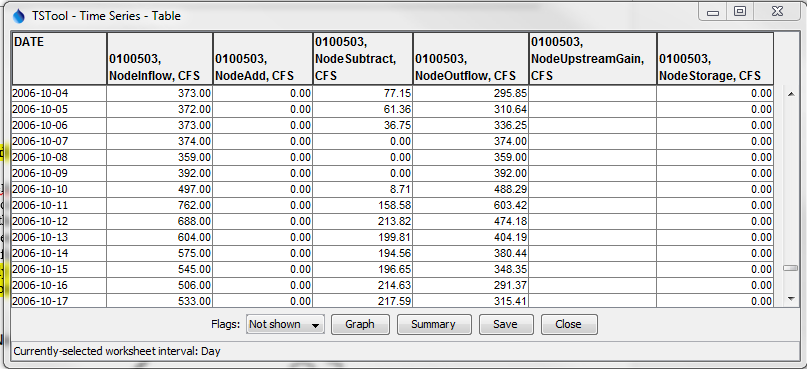
The command syntax is as follows:

AnalyzeNetworkPointFlow(Parameter=Value,…)

Command Parameters

| Parameter | Description | Default |
| --- | --- | --- |
| TableID | The identifier for the table defining the network. | None – must be specified. |
| NodeIDColumn | The name of the column in the network table containing node identifiers. Node identifiers will be used for the location ID part of time series identifiers. | None – must be specified. |
| NodeNameColumn | The name of the column in the network table containing node names. |  |
| NodeTypeColumn | The name of the column in the network table containing node types. The node type is used to specify what calculations will occur for the node. | None – must be specified. |
| NodeDistanceColumn | The name of the column in the network table containing node distance. The distance is the measure from the most downstream node and is used when GainMethod=Distance. | Must be specified when GainMethod=Distance. |
| NodeWeightColumn | The name of the column in the network table containing node weights, which is used to distribute gain/loss when GainMethod=Weight. | If not specified when GainMethod=Weight, gain/loss will be distributed evenly for the nodes. |
| DownstreamNodeIDColumn | The name of the column in the network table containing downstream node identifiers. This information defines the connectivity of the network. | None – must be specified. |
| NodeAddTypes | Node types for which time series are added to the node’s inflow to compute outflow, for example the Return node type in the above table example. The NodeTypeColumn table column is checked to determine the type for each node in the network. | No additions will occur. |
| NodeAddDataType | The time series data type to match for the node. The data type is used with the NodeID as the location ID to match available time series to use as input. This may be enhanced to allow a TSID pattern like %L-DivTotal, to allow more flexibility in matching time series. | No additions will occur. |
| NodeSubtractTypes | Node types for which time series are subtracted from the node’s inflow, for example the Diversion node type in the above table example. The NodeTypeColumn table column is checked to determine the type for each node in the network. | No subtractions will occur. |
| NodeSubtractDataType | The time series data type to match for the node. The data type is used with the NodeID as the location ID to match available time series to use as input. This may be enhanced to allow to a TSID pattern like %L-DivTotal, to allow more flexibility in matching time series. | No subtractions will occur. |
| NodeOutflowTypes | Node types for which time series outflows are set to the node’s time input time series, for example the Streamflow node type in the above table example. The NodeTypeColumn table column is checked to determine the type for each node in the network. | No known flows will be set – gain/loss cannot be computed. |
| NodeOutflowDataType | The time series data type to match for the node. The data type is used with the NodeID as the location ID to match available time series to use as input. This may be enhanced to allow a TSID pattern like %L-Streamflow, to allow more flexibility in matching time series. | No subtractions will occur. |
| NodeFlowThroughTypes | Node types for which time series outflows are set to the node’s inflow, for example the InstreamFlow node type in the above table example. The NodeTypeColumn table column is checked to determine the type for each node in the network. | No known flows will be set – gain/loss cannot be computed. |
| Interval | The time series interval to process. The interval is used with the node identifier and data type to match input time series. | None – must be specified. |
| AnalysisStart | The analysis start, which defines the period for output time series. Specify to the interval precision. | Global output period. |
| AnalysisEnd | The analysis end, which defines the period for output time series. Specify to the interval precision. | Global output period. |
| Units | Units for output time series. Warnings will be generated if input time series for the analysis are not consistent wit these units. |  |
| GainMethod | The method used to estimate gains between known point flow nodes:   * Distance – prorate the gain/loss between known point flows in a reach using the distance between nodes as compared to the total reach length. Currently this can be used only on non-branching networks. * None – no gain/loss is estimated, resulting in a discontinuity in an outflow jump above each known point flow. | None |
| OutputTable | The identifier for the output table to receive analysis results statistics. | No output table will be created. |

The following figure illustrates the output time series at a Diversion node type, for the case where GainMethod=None.



AnalyzeNetworkPointFLow\_OutuptTS

AnalyzeNetworkPointFlow() Node Output Time Series