Command Reference: ARMA()

Lag and Attenuate a Time Series Using AutoRegressive Moving Average

Version 06.08.02, 2004-08-02, Color, Acrobat Distiller

The `ARMA()` command lags and attenuates a time series (e.g., to route a streamflow time series downstream). This approach preserves the "mass" of the data. The general equation for ARMA is:

$$O_t = a_1 * O_{t-1} + a_2 * O_{t-2} + \ldots + a_p * O_{t-p} + b_0 * I_t + b_1 * I_{t-1} + \ldots + b_q * I_{t-q}$$

Where:

t = time step

O_t = output value at time t

I_t = input value at time t

a, b = ARMA coefficients

and the p and q values indicate the degree of the equation: ARMA(p,q).

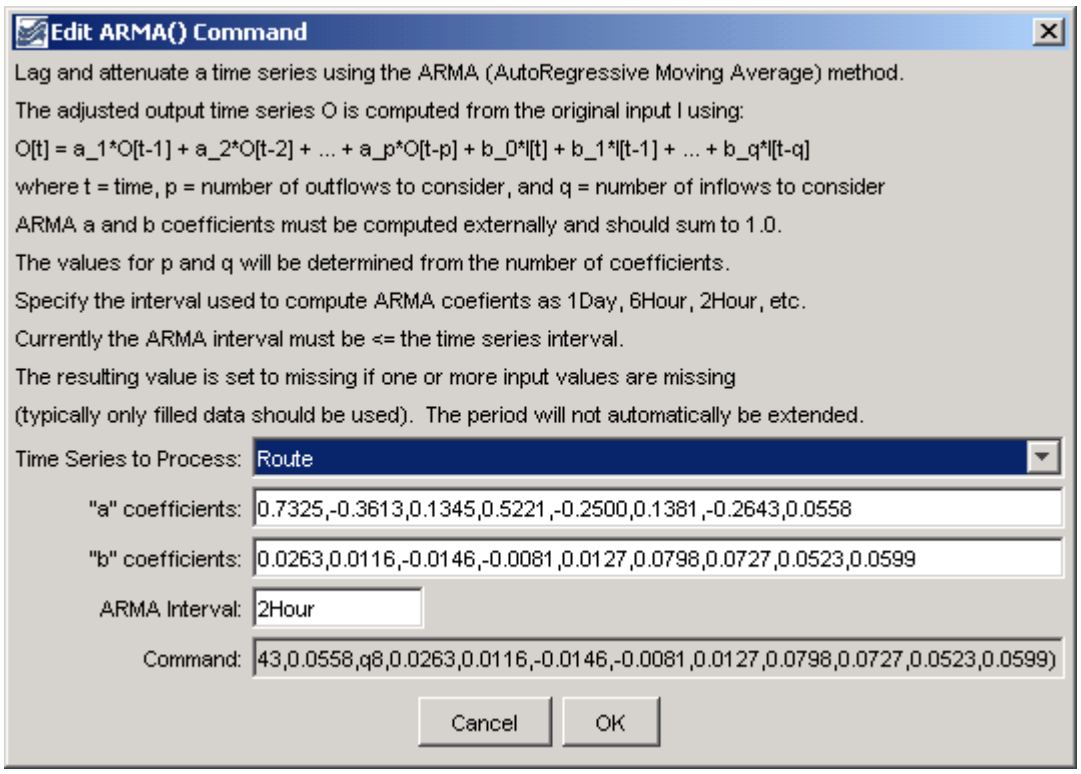
The ARMA coefficients are determined by analyzing historical data and may be developed using a data interval that is different than the data interval of the time series that is being manipulated. The coefficients are typically computed by an external analysis program (TSTool does not perform this function).

The time series to process can have any interval. The *a* and *b* coefficients are listed in the dialog from left-most to right-most in the equation. Note that there are p *a*-coefficients and $(q + 1)$ *b*-coefficients (because there is a *b*-coefficient at time t_0). The interval used to compute the ARMA coefficients can be different from the data interval but the data and ARMA intervals must be divisible by a common interval. The ARMA algorithm is executed as follows:

1. The data and ARMA intervals are checked and if they not the same, the data are expanded by duplicating each value into a temporary array. For example, if the data interval is 6Hour and the ARMA interval is 2Hour, each data value is expanded to three data values (2Hour values). If the data interval is 6Hour and the ARMA interval is 10Hour, each data value is expanded to three data values (2Hour values).
2. The ARMA equation is applied at each point in the expanded data array. However, because the ARMA coefficients were developed using a specific interval, only the data values at the ARMA interval are used in the equation. For example, if the expanded data array has 2Hour data and the ARMA interval is 10Hour, then every fifth value will be used (e.g., t corresponds to the "current" value and $t - 1$ corresponds to the fifth value before the current value). Because the ARMA algorithm depends on a number of previous terms in both the input and output, there will be missing terms at the beginning of the data array and in cases where missing data periods are encountered. Ideally ARMA will be applied to filled data and only the initial conditions will be an issue. In this case the output period should ideally be less than the total period so that the initial part of the routed time series can be ignored. In cases where *O* values are missing, the algorithm first tries to use the *I* values. If any values needed for the result are missing, the result is set to missing.
3. The final results are converted to a data interval that matches the original input, if necessary. If the original data interval and the ARMA interval are the same, no conversion is necessary. For example, if the original data interval is 6Hour and the ARMA interval is 10Hour, then the expanded data

interval will be 2Hour. Consequently, three sequential expanded values are averaged to obtain the final 6Hour time series.

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



ARMA

ARMA() Command Editor

The command syntax is as follows:

```
ARMA(TSID,ARMAInterval,pP,a1,…,aP,qQ,b0,…,bQ)
```

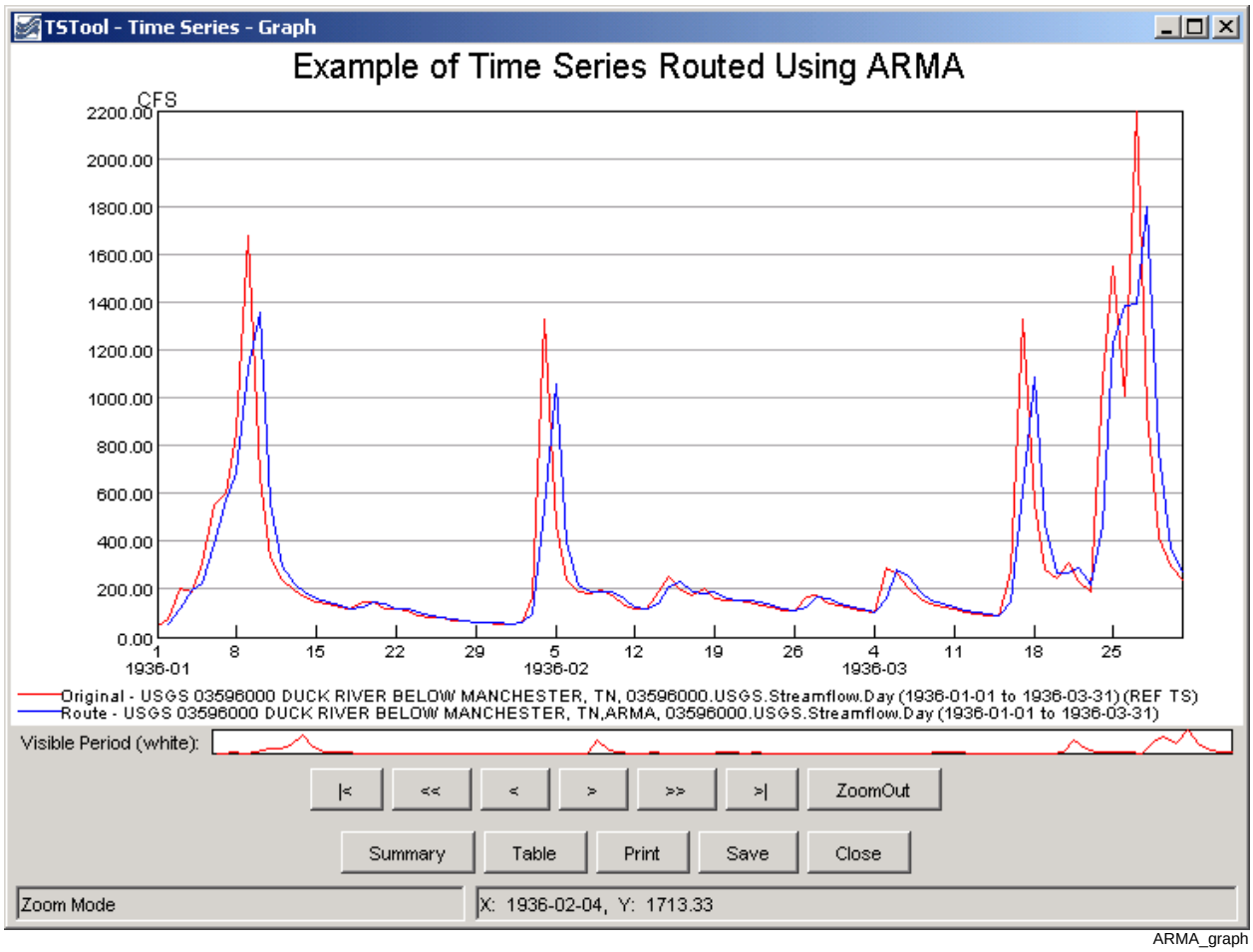
Command Parameters

Parameter	Description	Default
`TSID`	The time series identifier or alias for the time series to be modified.	None – must be specified.
`ARMAInterval`	The ARMA interval to use in the analysis	None – must be specified.
`pP`	Indicates the p-degree of the equation (`P` = number of *a* coefficients).	None – must be specified.
`a1,…,aP`	a coefficients.	None – must be specified.
`qQ`	Indicates the q-degree of the equation (`Q` = number of *b* coefficients - 1).	None – must be specified.
`b0,…,bQ`	b coefficients.	None – must be specified.

A sample commands file is as follows:

```
setOutputPeriod(1936-01-01,1936-03-31)
TS Original = readUsgsNwis("G03596000.in1",*,*)
TS Route = copy(Original)
# TS Route = readUsgsNwis("G03596000.in1",*,*)
ARMA(Route,2Hour,p8,0.7325,-0.3613,0.1345,0.5221,-0.2500,0.1381,
-0.2643,0.0558,q8,0.0263,0.0116,-0.0146,
-0.0081,0.0127,0.0798,0.0727,0.0523,0.0599)
writeDateValue(OutputFile="TS_ARMA.out")
```

The following figure shows the original and routed time series.

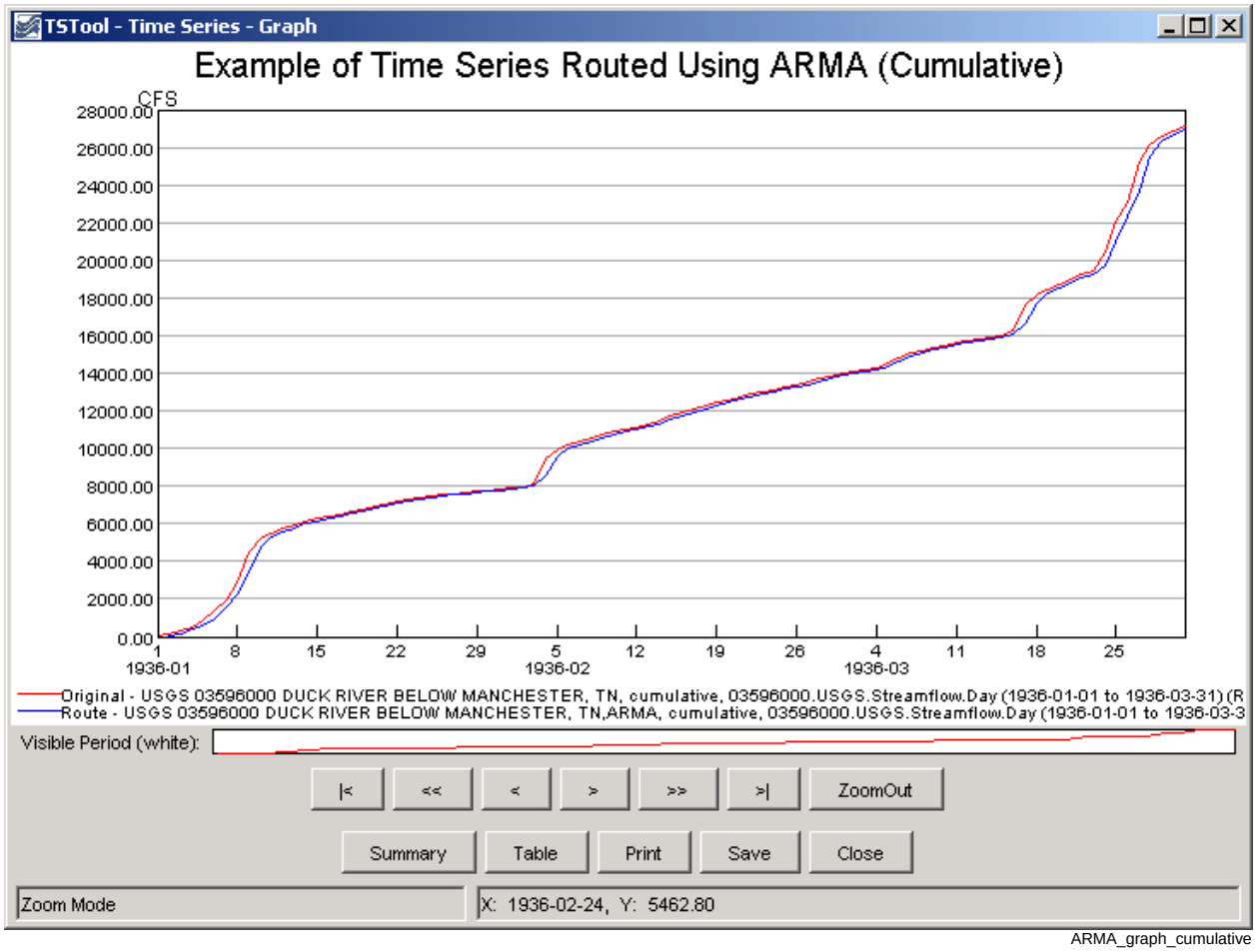


ARMA_graph

Example Graph Showing Original and ARMA-Routed Time Series

The `cumulate()` command can be used to verify mass balance of the original and routed time series (see the `cumulate()` command discussion below). For example, insert a `cumulate(*,CarryForwardIfMissing)` command near the end of a command file.

The following figure shows the time series from the previous graph, this time as cumulative time series.



ARMA_graph_cumulative

Example Graph Showing Original and ARMA-Routed Time Series as Cumulative Values