



## USING LUPROC

Effective:  
11/30/12

Revision:  
A

# Using LUPROC

---

## Overview

LUPROC is a MISERnet processor that uses point values as inputs to table lookups, which determine the values of result points. The lookup tables are text files and are stored in SITE\$DATA:\*.INTRP.

LUPROC performs interpolation on one or more points and writes the results to the database. Interpolation is a method of constructing new data points within the range of a discrete set of known data points. Interpolation often involves a number of data points, obtained by sampling or experimentation, which represent the values of a function for a limited number of values of the independent variable.

LUPROC is Change-Of-State (COS) driven. Whenever a COS occurs on a point that is an input to a lookup table, LUPROC immediately calculates the value of the result point. Result points must be INPUT SUBTYPE CALC.

## Lookup Table Files

Each lookup table file must include these sections:

- `_ROUNDING:` — occurs once.
- `_FUNCTION:` — occurs 1+ times.
- `_DATA:` — occurs once.
- `_END_DATA:` — occurs once.

## Rounding

The `_ROUNDING:` entry occurs on one line and has one or more rounding rule keyword(s) separated by commas. There must be as many rounding rule keywords as there are inputs in a `_DATA:` line. There are four ways to determine the type of rounding to use to obtain the proper value. The rounding rule keywords are:

- `UP` — this rounds the actual value up to next value in the lookup table.
- `DOWN` — this rounds the actual value down to the next value in the lookup table.
- `INTERPOLATE` — this uses a calculation to determine the value based on the two values that the actual value falls between.
- `NEAREST` — this rounds the actual value to nearest value in the lookup table.

## USING LUPROC

### Function

Each `_FUNCTION:` entry occurs on just one line and has (in this order):

- One or more input point acronym(s) separated by commas.
- A colon.
- A result point acronym.

There must be as many input acronyms as there are inputs in a `_DATA:` line. There can be more than one `_FUNCTION:` line entry, so there can be more than one set of points using the table data.

### Data and End Data

Between `_DATA:` and `_END_DATA:` are decimal numbers comprising the table data. Each line in this section has (in this order):

- One or more input number(s) separated by commas.
- A colon.
- A result value.

Each line must have the same number of input values, and one result.

### Example 1

```
!           TURBINE LEAKAGE
!           (UNITS IN DSF)
!
!_ROUNDING: DOWN
!_FUNCTION: B-PWR-FLOW: B-TURB-LEAK
!_DATA:
0.0: 20
301.0: 18
901.0: 16
1501.0: 14
2101.0: 12
2701.0: 10
3301.0: 8
3901.0: 6
4501.0: 4
5101.0: 2
5701.0: 0
!_END_DATA:
```

If the B-PWR-FLOW point reports a COS to 2315.7, the B-TURB-LEAK calculation will be:

1. As the file is written with `ROUNDING: DOWN`, the input used is 2101.0 and the result is 12.
2. If `_ROUNDING:UP` were used, the input would be 2701.0 and the result would be 10.
3. If `_ROUNDING:INTERPOLATE` were used, the input would actually be 2315.7 and the result would be the interpolation between 12 and 10 which would be 11.284.
4. If `_ROUNDING:NEAREST` were used, the input would be 2101.0 and the result would be 12.

## Example 2

```

!           TURBINE DISCHARGE TABLE
!
!_ROUNDING: DOWN, INTERPOLATE, INTERPOLATE
!_FUNCTION: Q.GVALVE, Q.GMW, Q.GHEAD: Q.GFLOW
!_FUNCTION: K.G2VALVE, K.G2MW, K.HEAD: K.G2FLOW
!_FUNCTION: K.G3VALVE, K.G3MW, K.HEAD: K.G3FLOW
!_FUNCTION: K.G4VALVE, K.G4MW, K.HEAD: K.G4FLOW
!_FUNCTION: K.G5VALVE, K.G5MW, K.HEAD: K.G5FLOW
!_FUNCTION: K.G6VALVE, K.G6MW, K.HEAD: K.G6FLOW
!_FUNCTION: K.G7VALVE, K.G7MW, K.HEAD: K.G7FLOW
!_DATA:
0 , 0 , 0 : 0.0
0 , 0 , 78 : 0.0
0 , 0 , 80 : 0.0
0 , 0 , 82 : 0.0
0 , 0 , 84 : 0.0
0 , 0 , 86 : 0.0
0 , 0 , 88 : 0.0
0 , 0 , 90 : 0.0
0 , 0 , 92 : 0.0
0 , 0 , 94 : 0.0
0 , 0 , 96 : 0.0
0 , 0 , 98 : 0.0
0 , 0 , 100 : 0.0
0 , 0 , 102 : 0.0
0 , 0 , 104 : 0.0
0 , 0 , 106 : 0.0
0 , 0 , 108 : 0.0
~~~~~
1 , 59 , 78 : 17882.8707733934
1 , 59 , 80 : 17402.1739102817
1 , 59 , 82 : 16413.919990124
1 , 59 , 84 : 14620.2372382931
1 , 59 , 86 : 13336.2920137249
1 , 59 , 88 : 12697.926173768
1 , 59 , 90 : 11695.8166402042
1 , 59 , 92 : 10871.8794464427
1 , 59 , 94 : 10245.7935576799
1 , 59 , 96 : 9739.10735798738
1 , 59 , 98 : 9276.81402453452
1 , 59 , 100 : 8837.44763267783
1 , 59 , 102 : 8492.41788529461
1 , 59 , 104 : 8155.88380548972
1 , 59 , 106 : 7846.28496814357
1 , 59 , 108 : 7589.85299967296
!_END_DATA:

```

The above example shows a combination of points and rounding types.

There are three inputs to each table lookup. Seven sets of points use the data table. If point Q.VALVE reports 1 while Q.GMW is 59 and Q.GHEAD is 78, then Q.GFLOW is calculated as 17882.8707733934.

**USING LUPROC**

## Lookup Table Example

	Generation (MW)																															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
64.7	546	718	889	1,061	1,232	1,404	1,575	1,747	1,918	2,090	2,261	2,433	2,614	2,815	3,015	3,217	3,418	3,620	3,822	4,024	4,226	4,429	4,631	4,834	5,037							
64.8	547	718	889	1,060	1,231	1,402	1,573	1,744	1,915	2,086	2,257	2,428	2,609	2,810	3,010	3,211	3,412	3,613	3,815	4,016	4,217	4,419	4,621	4,823	5,044							
64.9	549	719	890	1,060	1,231	1,401	1,572	1,742	1,913	2,083	2,254	2,424	2,605	2,805	3,005	3,206	3,406	3,607	3,807	4,008	4,209	4,410	4,610	4,811	5,032							
65.0	550	720	890	1,060	1,230	1,400	1,570	1,740	1,910	2,080	2,250	2,420	2,600	2,800	3,000	3,200	3,400	3,600	3,800	4,000	4,200	4,400	4,600	4,800	5,020	5,260	5,500	5,770	6,080	6,380		
65.1	549	719	889	1,058	1,228	1,398	1,568	1,738	1,907	2,077	2,247	2,417	2,596	2,796	2,996	3,195	3,395	3,594	3,794	3,994	4,193	4,393	4,593	4,793	5,012	5,251	5,490	5,759	6,067	6,365		
65.2	548	718	887	1,057	1,226	1,396	1,566	1,735	1,905	2,074	2,244	2,414	2,593	2,792	2,991	3,190	3,390	3,589	3,788	3,987	4,186	4,386	4,586	4,785	5,004	5,242	5,480	5,748	6,054	6,350		
65.3	547	716	886	1,055	1,225	1,394	1,563	1,733	1,902	2,072	2,241	2,410	2,589	2,788	2,987	3,186	3,384	3,583	3,782	3,981	4,180	4,379	4,578	4,778	4,996	5,233	5,471	5,737	6,041	6,336		
65.4	546	715	884	1,054	1,223	1,392	1,561	1,730	1,900	2,069	2,238	2,407	2,586	2,784	2,982	3,181	3,379	3,578	3,776	3,974	4,173	4,372	4,571	4,770	4,988	5,224	5,461	5,726	6,028	6,321		
65.5	545	714	883	1,052	1,221	1,390	1,559	1,728	1,897	2,066	2,235	2,404	2,582	2,780	2,978	3,176	3,374	3,572	3,770	3,968	4,166	4,365	4,564	4,763	4,980	5,215	5,451	5,715	6,015	6,306		
65.6	544	713	882	1,050	1,219	1,388	1,557	1,726	1,894	2,063	2,232	2,401	2,578	2,776	2,974	3,171	3,369	3,566	3,764	3,962	4,159	4,358	4,557	4,756	4,972	5,206	5,441	5,704	6,002	6,291		
65.7	543	712	880	1,049	1,217	1,386	1,555	1,723	1,892	2,060	2,229	2,398	2,575	2,772	2,969	3,166	3,364	3,561	3,758	3,955	4,152	4,351	4,550	4,748	4,964	5,197	5,431	5,693	5,989	6,276		
65.8	542	710	879	1,047	1,216	1,384	1,552	1,721	1,889	2,058	2,226	2,394	2,571	2,768	2,965	3,162	3,358	3,555	3,752	3,949	4,146	4,344	4,542	4,741	4,956	5,188	5,422	5,682	5,976	6,262		
65.9	541	709	877	1,046	1,214	1,382	1,550	1,718	1,887	2,055	2,223	2,391	2,568	2,764	2,960	3,157	3,353	3,550	3,746	3,942	4,139	4,337	4,535	4,733	4,948	5,179	5,412	5,671	5,963	6,247		
66.0	540	708	876	1,044	1,212	1,380	1,548	1,716	1,884	2,052	2,220	2,388	2,564	2,760	2,956	3,152	3,348	3,544	3,740	3,936	4,132	4,330	4,528	4,726	4,940	5,170	5,402	5,660	5,950	6,232		
66.1	539	707	875	1,042	1,210	1,378	1,546	1,714	1,881	2,049	2,217	2,385	2,560	2,756	2,952	3,147	3,343	3,538	3,734	3,930	4,125	4,323	4,521	4,719	4,932	5,161	5,392	5,649	5,937	6,217		
66.2	538	706	873	1,041	1,208	1,376	1,544	1,711	1,879	2,046	2,214	2,382	2,557	2,752	2,947	3,142	3,338	3,533	3,728	3,923	4,118	4,316	4,514	4,711	4,924	5,152	5,382	5,638	5,924	6,202		
66.3	537	704	872	1,039	1,207	1,374	1,541	1,709	1,876	2,044	2,211	2,378	2,553	2,748	2,943	3,138	3,332	3,527	3,722	3,917	4,112	4,309	4,506	4,704	4,916	5,143	5,373	5,627	5,911	6,188		
66.4	536	703	870	1,038	1,205	1,372	1,539	1,706	1,874	2,041	2,208	2,375	2,550	2,744	2,938	3,133	3,327	3,522	3,716	3,910	4,105	4,302	4,499	4,696	4,908	5,134	5,363	5,616	5,898	6,173		
66.5	535	702	869	1,036	1,203	1,370	1,537	1,704	1,871	2,038	2,205	2,372	2,546	2,740	2,934	3,128	3,322	3,516	3,710	3,904	4,098	4,295	4,492	4,689	4,900	5,125	5,353	5,605	5,885	6,158		
66.6	534	701	868	1,034	1,201	1,368	1,535	1,702	1,868	2,035	2,202	2,369	2,542	2,736	2,930	3,123	3,317	3,510	3,704	3,898	4,091	4,288	4,485	4,682	4,892	5,116	5,343	5,594	5,872	6,143		
66.7	533	700	866	1,033	1,199	1,366	1,533	1,699	1,866	2,032	2,199	2,366	2,539	2,732	2,925	3,118	3,312	3,505	3,698	3,891	4,084	4,281	4,478	4,674	4,884	5,107	5,333	5,583	5,859	6,128		
66.8	532	698	865	1,031	1,198	1,364	1,530	1,697	1,863	2,030	2,196	2,362	2,535	2,728	2,921	3,114	3,306	3,499	3,692	3,885	4,078	4,274	4,470	4,667	4,876	5,098	5,324	5,572	5,846	6,114		
66.9	531	697	863	1,030	1,196	1,362	1,528	1,694	1,861	2,027	2,193	2,359	2,532	2,724	2,916	3,109	3,301	3,494	3,686	3,878	4,071	4,267	4,463	4,659	4,868	5,089	5,314	5,561	5,833	6,099		

The table above is an example of a spreadsheet for a lookup table that uses water level and electrical power generation to determine water flow. The X axis is the value of point that is the water level in feet. The Y axis is the point that measures the megawatts generated. LUPROC uses the intersection of the two points to get a result of the water flow value. A DCL procedure was used to translate this into the file: C\_TURB\_1\_FLOW.INTRP, as shown below.

## USING LUPROC

```
!           Turbine Discharge Table
!
!_ROUNDING: NEAREST, DOWN
!_FUNCTION: C-NETHEAD, C-U1-MWH-O: C-T1-MINFLOW
!_DATA:
59.0, 0 : 0.0
59.0, 1 : 470.0
59.0, 2 : 670.0
...
64.6, 23 : 4641.0
64.6, 24 : 4845.0
64.6, 25 : 5069.0
64.7, 0 : 0.0
64.7, 1 : 546.0
64.7, 2 : 718.0
64.7, 3 : 889.0
64.7, 4 : 1061.0
64.7, 5 : 1232.0
64.7, 6 : 1404.0
64.7, 7 : 1575.0
64.7, 8 : 1747.0
64.7, 9 : 1918.0
64.7, 10 : 2090.0
64.7, 11 : 2261.0
64.7, 12 : 2433.0
64.7, 13 : 2614.0
64.7, 14 : 2815.0
64.7, 15 : 3015.0
64.7, 16 : 3217.0
64.7, 17 : 3418.0
64.7, 18 : 3620.0
64.7, 19 : 3822.0
64.7, 20 : 4024.0
64.7, 21 : 4226.0
64.7, 22 : 4429.0
64.7, 23 : 4631.0
64.7, 24 : 4834.0
64.7, 25 : 5057.0
64.8, 0 : 0.0
64.8, 1 : 547.0
...
```

## **Restrictions**

LUPROC is particular about the ordering of the table data. The first input must be in increasing order from top to bottom. The second input must be increasing order from top to bottom within one value of the first input, etc. This is why in example 2, all the entries for 0, 0, n are together with the n's in increasing order. Next would come 0, 1, n and so on.

If LUPROC is run on a workstation (not the "A" or "B" Host), then the lookup file(s) that workstation uses can only exist on that workstation (at least with the .INTRP file extension). In other words, there is nothing in LUPROC to prevent it from trying to do calculations that are supposed to be done in a different workstation.

LUPROC does not handle acronyms longer than 14 characters.

## USING LUPROC

### Running LUPROC on a MISER Computer

There must be a site specific executable: SITE\$EXE:LUPROC.EXE. The following entry must exist in SITE\$DATA:START\_<nodename>.DAT:

```
RUN LUPROC -  
  /DELAY="0 00:03:00.00 -  
  /PROCESS_NAME=LUPROC -  
  /PRIORITY=4
```