

Table-driven modeling in Modelica

The Modelica Language Reference Manual [1] does not describe any pre-defined functions for table-driven modeling. The Modelica Standard Library 3.1 [2] contains two packages (also called libraries) that provide features for table driven modeling.

Modelica.Math

Modelica.Math is a library of mathematical functions (e.g., sin, cos) and of functions operating on vectors and matrices. Modelica.Math provides two deprecated functions for interpolation.

Name	Description
tempInterpol1	Temporary function for linear interpolation (will be removed)
tempInterpol2	Temporary function for vectorized linear interpolation (will be removed)

In case of tempInterpol1, the first column of the table that is used by the functions describes the independent variable. Each other column describes the dependent values for another map. The first function assigns a scalar output value to the scalar input. The function is selected by the index of the column to be interpolated.

The second function tempInterpol2 assigns an array of output values to the scalar input. The first column of the table describes the independent variable. The functions are selected by the indices of the columns to be interpolated.

Modelica.Blocks.Tables

Modelica.Blocks.Tables is a library of blocks to interpolate in one and two-dimensional tables. It provides three blocks for this purpose.

Name	Description
CombiTable1D	Table look-up in one dimension (matrix/file) with n inputs and n outputs
CombiTable1Ds	Table look-up in one dimension (matrix/file) with one input and n outputs
CombiTable2D	Table look-up in two dimensions (matrix/file)

CombiTable1D

In [2] the following description is given: “**Linear interpolation** in **one** dimension of a **table**. Via parameter **columns** it can be defined how many columns of the table are interpolated. If, e.g., columns={2,4}, it is assumed that 2 input and 2 output signals are present and that the first output interpolates the first input via column 2 and the second output interpolates the second input via column 4 of the table matrix. The grid points and function values are stored in a matrix "table[i,j]", where the first column "table[:,1]" contains the grid points and the other columns contain the data to be interpolated.

Example:

```
table = [0, 0;
        1, 1;
        2, 4;
        4, 16]
```

```
If, e.g., the input u = 1.0, the output y = 1.0,
e.g., the input u = 1.5, the output y = 2.5,
e.g., the input u = 2.0, the output y = 4.0,
e.g., the input u = -1.0, the output y = -1.0 (i.e. extrapolation).
```

[...]

If tables are read from an ASCII-file, the file need to have the following structure ("-----" is not part of the file content):

```
-----
#1
double tab1(5,2)    # comment line
 0   0
 1   1
 2   4
 3   9
 4  16
double tab2(5,2)    # another comment line
 0   0
 2   2
 4   8
 6  18
 8  32
-----
```

Note, that the first two characters in the file need to be "#1". Afterwards, the corresponding matrix has to be declared with type, name and actual dimensions. Finally, in successive rows of the file, the elements of the matrix have to be given. Several matrices may be defined one after another.“

*CombiTable1D*s

In [2] the following description is given: “**Linear interpolation** in **one** dimension of a **table**. Via parameter **columns** it can be defined how many columns of the table are interpolated. If, e.g., `icol={2,4}`, it is assumed that one input and 2 output signals are present and that the first output interpolates via column 2 and the second output interpolates via column 4 of the table matrix.”

CombiTable2D

In [2] the following description is given: “**Linear interpolation** in **two** dimensions of a **table**. The grid points and function values are stored in a matrix "table[i,j]", where:

- the first column "table[2:,1]" contains the u[1] grid points,
- the first row "table[1,2:]" contains the u[2] grid points,
- the other rows and columns contain the data to be interpolated.

Example:

```

      |   |   |   |
      | 1.0 | 2.0 | 3.0 | // u2
-----*-----*-----*-----*
1.0 | 1.0 | 3.0 | 5.0 |
-----*-----*-----*-----*
2.0 | 2.0 | 4.0 | 6.0 |
-----*-----*-----*-----*
// u1

is defined as
table = [0.0, 1.0, 2.0, 3.0;
        1.0, 1.0, 3.0, 5.0;
        2.0, 2.0, 4.0, 6.0]
If, e.g. the input u is [1.0;1.0], the output y is 1.0,
e.g. the input u is [2.0;1.5], the output y is 3.0.
```

[...]“

The table matrix can be defined

- as two-dimensional array
 - In the case of the interpolation in one dimension the first column contains the values of the independent variable. The columns with the dependent variables can be selected by parameters of the 1D functions
 - In the case of the interpolation in two dimensions the first column and the first row describe the grid points. Table contents at position `table[1,1]` has no meaning. All other table elements describe function values
- as contents of a file
 - ASCII files and
 - binary files in Matlab 4 storage format are supported
- by a C function `usertab`

Parameters describe the applied methods for

- Smoothness
 - `LinearSegments` "Table points are linearly interpolated",
 - `ContinuousDerivative` "Table points are interpolated such that the first derivative is continuous"
- Extrapolation
 - `HoldLastPoint` "Hold the last table point outside of the table scope"
 - `LastTwoPoints` "Extrapolate linearly through the last two table points outside the scope of the table"
 - `Periodic` "Repeat the table scope periodically"

References

- [1] Modelica® - A Unified Object-Oriented Language for Physical Systems Modeling Language Specification Version 3.2. Modelica Association, March 24, 2010.
Online: <http://www.modelica.org/documents/ModelicaSpec32.pdf>
- [2] Modelica Standard Library 3.1_built6. Released January 17, 2010.
Online: <http://www.modelica.org/libraries/Modelica>
Access: 3.1-build-6/Modelica%203.1/help/Modelica.html

Contact:

Joachim Haase

E-Mail: joachim.haase@eas.iis.fraunhofer.de