Using Functions Based on Interpolated Data

To define functions based on interpolated data, use the **Functions** dialog box, which you open from the **Options** menu. To define a new interpolation function, click the **New** button to open the **New Function** dialog box, specify a name for the function and click the **Interpolation** button. Then select a method for entering data, a data source, if applicable, and press **OK**.

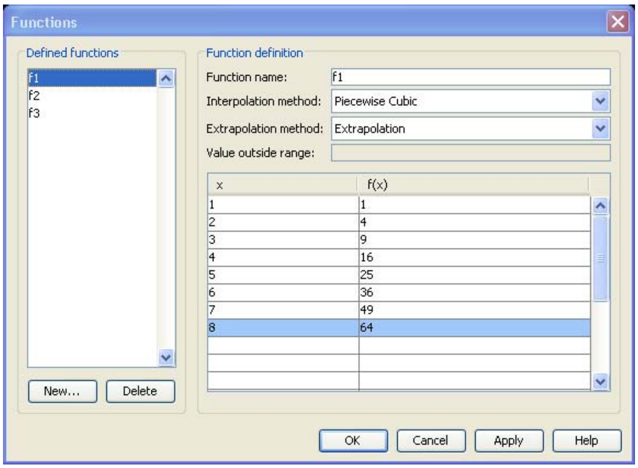

Figure 4-3: The Functions dialog box, editing an interpolation function.

For functions of one variable, you can select between the following interpolation methods:

- Nearest neighbor
- Linear
- Piecewise cubic
- Cubic spline

For functions of more than one variable COMSOL Multiphysics only supports the nearest neighbor and linear interpolations. Piecewise cubic interpolation is a method using a piecewise cubic Hermite polynomial with continuous first derivatives. It preserves the shape of the data and respects monotonicity. The cubic spline method also performs interpolation with a piecewise cubic polynomial. Here, even second derivatives are continuous; however, monotonicity is not necessarily respected.

Also select what COMSOL Multiphysics should do when arguments fall outside the grid. There are three extrapolation methods to choose from:

- Constant
- Extrapolation
- Specific Number

Constant means that the value from the closest point inside the grid will be used while Extrapolation evaluates the polynomial from the closest grid point at the actual point where a value is requested. If you select Specific Number, you can assign a single value, usually zero or NaN, to all points outside the grid.

For functions of one variable you can enter the data directly into a lookup table. For functions of one to three variables (which often correspond to the space coordinates), you can either retrieve data from a text file, COMSOL Script or MATLAB. The file should have the same format as the file produced when exporting on the format **Grid, data** on a regular grid using **Export>Postprocessing Data** on the **File** menu:

```
% Grid
x grid points separated by spaces
y grid points separated by spaces (optional)
z grid points separated by spaces (optional)
% Data
Data values separated by spaces
```

Each row contains values for different x grid points for a fixed y and z. The rows first increase the y grid value and then the z grid value. The grid points can also represent another independent variable that the data values depend on. For example, the "grid points" can be temperature values and the data values the thermal conductivity at these temperatures (see the example in "Interpolation of Measured Data" on page 211). It is important to use a comment line starting with `%` to separate the grid points or other interpolation points and the data values that are associated with these coordinates or interpolation points.

To retrieve data from COMSOL Script or MATLAB, create a structure on the command line with the fields `x`, `y` (optional), `z` (optional), and `data`. `x`, `y`, and `z` are vectors specifying the grid, and `data` is a multidimensional array of the same format as produced by the `meshgrid` function.

When COMSOL Multiphysics evaluates the function, the software performs a structured interpolation from the data values on the grid to the coordinates where the function is evaluated. See the model "Rock Fracture Flow" in the *COMSOL Multiphysics Model Library* for an example using data interpolation.

Interpolation of Measured Data

Many material properties and other input data are not given by an analytical expression but can only be obtained by measurements. You can then use these measurements to approximate the material properties via interpolation during a COMSOL Multiphysics analysis.

The measured data can depend on the solution variables. A typical example of this is when approximating hysteresis curves for magnetic materials or temperature-dependent thermal properties such as thermal conductivity. Interpolation is usually made in one dimension, that is, the material property is considered a function of one variable.

The easiest way to use interpolation of measured data is to create an interpolation function (lookup table) directly. See "Using Functions Based on Interpolated Data" on page 155 for details. If you use COMSOL Script or MATLAB, you can also use interpolation functions in a COMSOL Script or MATLAB function. The model "A Rock Fracture Flow Model" on page 246 in the *COMSOL Multiphysics Model Library* uses both methods.

Modeling Thermal Conductivity from Measurements

To illustrate the use of interpolation for nonlinear materials, create a model with a thermal conductivity that is based on measurements at different temperatures.

SETTING UP THE INTERPOLATION ROUTINE

The first step in setting up the interpolation routine is to define the values for the measured data. You can also load the data from a text file. The following table shows the measured data:

TABLE 4-12: THERMAL CONDUCTIVITY

TEMPERATURE	THERMAL CONDUCTIVITY
20	1.4
22	3.6
24	6.3
26	4.7
28	4.3

TABLE 4-12: THERMAL CONDUCTIVITY

TEMPERATURE	THERMAL CONDUCTIVITY
30	3.7
32	3.6
34	4.8
36	5.2
38	5.1
40	4.8

You can enter this data set directly into the **Functions** dialog box:

MODELING USING THE GRAPHICAL USER INTERFACE—LOOKUP TABLE

1 Start COMSOL Multiphysics.
2 Select **2D** from the **Space dimension** list and **COMSOL Multiphysics>Heat Transfer>Conduction>Steady-state analysis** in the list of application modes. Click **OK**.
3 Draw a square with corners at (0, 0) and (1, 1).
4 Click the **Zoom Extents** toolbar button to adjust the coordinate system to the size of the rectangle.
5 Open the **Boundary Settings** dialog box. Select all boundaries and the select **Temperature** in the **Boundary condition** list. Type `20` in the **Temperature** edit field and click **OK**.

   Define the interpolation function `therm_cond1(T)` by directly entering the conductivity data in Table 4-12:

6 From the **Options** menu, choose **Functions**.
7 Click the **New** button.
8 In the **New Function** dialog box, type `therm_cond1` in the **Function name** edit field and click the **Interpolation** button. Click **OK**.

9 In the table, type the values for temperature and the thermal conductivity in the **x** and **f(x)** columns, respectively.

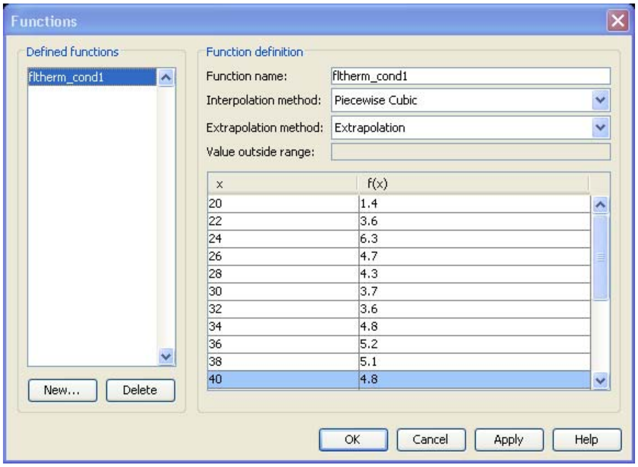

Figure 4-20: Interpolation function definition using a table in the Functions dialog box.

10 Click **OK**.

Next, use this data in the specification of the thermal conductivity:

11 Open the **Subdomain Settings** dialog box and type `therm_cond1(T)` in the **Thermal conductivity** edit field. Type `750` in the **Heat source** edit field.

12 Click the **Init** tab and then type `20` in the **Temperature** edit field to set the initial value. Click **OK**.

13 Click the **Initialize Mesh** button to create a mesh.

14 Solve the problem by clicking the **Solve** button.

MODELING USING THE GRAPHICAL USER INTERFACE—DATA FROM FILE

In this case, the same interpolation data is available as a text file, `thermcond.txt`, with the following format:

```
% Temperatures
20 22 24 26 28 30 32 34 36 38 40
% Thermal conductivity
1.4 3.6 6.3 4.7 4.3 3.7 3.6 4.8 5.2 5.1 4.8
```

Store a text file with this context as `thermcond.txt` somewhere on your file system before using the following step to use the interpolation data from a file:

1 Follow steps 1–7 in the previous section

2 In the **New Function** dialog box, type `therm_cond1` in the **Function name** edit field and click the **Interpolation** button. Select **File** in the **Use data from** list.
3 Click the **Browse** button and select the file `thermcond.txt` in the location where you saved it.
4 Click **OK** to close the **New Function** dialog box.
5 In the **Functions** dialog box, select **Extrapolation** in the **Extrapolation method** list to use extrapolation of the thermal conductivity outside of the specified temperature range. Click **OK**.
6 Open the **Subdomain Settings** dialog box and type `therm_cond1(T)` in the **Thermal conductivity** edit field. Type `750` in the **Heat source** edit field.
7 Click the **Init** tab and then type `20` in the **Temperature** edit field to set the initial value. Click **OK**.
8 Click the **Initialize Mesh** button to create a mesh.
9 Solve the problem by clicking the **Solve** button

USING A POLYNOMIAL TO DESCRIBE THE DATA

If possible, a preferred method is often to fit a polynomial to your data. Once you have an analytical function for your nonlinear material, for example, $T = a + bT + cT^2$, you can enter the function directly using the edit field for the conductivity. Alternatively you can define the expression in the expression list. In that case, COMSOL Multiphysics takes care of the differentiation for the Jacobian contribution.