Cubic Spline Interpolation in Verilog-AMS

Table models in Verilog-AMS [1]

The Verilog-AMS LRM promises the interpolation of maps of the following form

$$f : T \subset R^N \to R^M$$

The LRM describes the requirements concerning the data as follows: “A table of M dependent variables of dimension N are laid out in N+M columns in the file, with the independent variables appearing in the first N columns followed by the dependent variables in the remaining M columns. The independent variables are ordered from the outermost (slowest changing) variable to the innermost (fastest changing) variable. Though an isoline ordinate does not change for a given isoline, in this scheme the ordinate is repeated for each point of that isoline (thus keeping the input data as a set of data rows all with the same number of points). The result is a sequential listing of each isoline with the total number of points in the listing being equal to the total number of samples on all isolines.” Furthermore, how to handle the situation where more columns are provided than needed, how to interpolate between data points and how to extrapolate if a point is outside $T \subset R^N$ is described in [1]

An example of an ordered set of data points (where N=2 and M=1) is given ([1], p.228f)

```
# 2-D table model sample example
##
y    x  f(x,y)
#y=0 isoline
0.0 1.0 0.5
0.0 2.0 1.0
0.0 3.0 1.5
0.0 4.0 2.0
0.0 5.0 2.5
0.0 6.0 3.0
#y=0.5 isoline
0.5 1.0 1.0
0.5 3.0 2.0
0.5 5.0 3.0
#y=1.0 isoline
1.0 1.0 1.5
1.0 2.0 2.0
1.0 4.0 3.0
```

The Verilog-AMS LRM addresses the following approaches for the interpolation the values of a column (table_interp_char)

- I - Ignore the column
- C - Closest point lookup
- 1 - Linear interpolation
- 2 - Quadratic spline interpolation
- 3 - Cubic spline interpolation

Extrapolation of values is described by (table_extrap_char)

- C - Constant extrapolation
- L - Linear extrapolation
- E - Error

When formulating cubic spline the method depends on the extrapolation value (table_extrap_char) as follows

- C - the end point derivatives are set to zero
- L - natural spline,

Methods described by Dahlquist and Björck [2]

The Verilog-AMS LRM [1], section 9.20, points to [2] as reference for table interpolation methods. Chapter 4 of [2] studies several methods for univariate (N=1) interpolation and approximation methods. In the described methods M equals 1. Algorithms for rational and multidimensional interpolation (N > 1) are briefly surveyed.

Spline functions are described in section 4.4.2. They are defined on a grid

$$\Delta = \{a = x_0 < x_1 < \ldots < x_m = b\}$$

It is required that for a spline function *s(x)* of degree *k* the first *k-2* derivatives are continuous on [*a,b*]. For $x \in [x_{i-1}, x_i), i = 1,\ldots,m$ *s(x)* is a polynomial of degree <*k*.

Linear spline interpolating and cubic spline functions are explained in detail. It is mentioned that cubic spline functions which interpolate a given function on the grid Δ and have continuous first and second order derivatives are by far the most important [2, p. 419].

Closest point lookup (also know as nearest neigbor interpolation) and quadratic spline interpolation are not discussed in detail. An extensive representation of B-splines is given, see also [3].

Linear Splines (p. 419)

Linear spline interpolates the given values $y_i = f(x_i)$ by

$$s(x) = y_{i-1} + d_i \cdot (x - x_{i-1}) \text{ for } x \in [x_{i-1}, x_i), i = 1,\ldots,m \qquad \text{with } d_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}} .$$

Linear spline preserves monotonicity and convexity of the interpolated function. It has a discontinuous first derivative at the knots. This is a problem in many applications.

Cubic Splines (p. 420 ff)

The cubic spline interpolates the given values $y_i = f(x_i)$ by

$$s(x) = \theta y_i + (1-\theta) y_{i-1} + h_i \theta (1-\theta) [(k_{i-1} - d_i)(1-\theta) - (k_i - d_i)\theta] \text{ for } x \in [x_{i-1}, x_i), i = 1,\ldots,m$$

with $\theta = \frac{x - x_{i-1}}{x_i - x_{i-1}} \in [0,1)$, $h_i = x_i - x_{i-1}$ and $d_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$.

The constants $k_0,\ldots,k_m$ depend on all data points. *m-1* conditions are given by the requirement that the second order derivatives of the cubic spline at $x_1,\ldots,x_{m-1}$ are continuous. Thus, two additional equations are necessary to complete the conditions. Special cases are

Complete cubic spline interpolant

In this case the derivatives at the end points are known. The additional conditions are given by:

$$k_0 = f'(a) = s'(x_0) \quad \text{and} \quad k_m = f'(b) = \lim_{x->x_m} s'(x)$$

Natural spline interpolant

In this case, the spline is straight outside the interval [*a, b*]. The second order derivatives at the boundary points are zero. The additional conditions are given by

$$0 = f''(a) = s''(x_0) \quad \text{and} \quad 0 = f''(b) = \lim_{x->x_m} s''(x)$$

Multidimensional interpolation (p. 395f)

It is figured out in [2] that much of the theory for univariate interpolation can be generalized to multidimensional interpolation problems. It is provided that the function is specified on a Cartesian (tensor) grid. A simple example that uses a function of two dimensions is presented. However, it is established that the extension to more dimensions is not difficult.

Assume that the given function values are

$$f_{ij} = f(x_i, y_j) \quad \text{where } i = 1,\ldots,n \quad \text{and} \quad j = 1,\ldots,m \,.$$

We want to compute the function value for given arguments *x* and *y*. At first, for each y_j ($j = 1,\ldots,m$) we use univariate interpolation to approximate $f(x, y_j)$. Next, these *m* values are used for univarite interpolation to approximate $f(x, y)$. The same result should be obtained, whether the interpolation first is done for *x* and then for *y* or vice versa.

References

[1] Verilog-AMS Language Reference Manual – Version 2.3.1. Accellera, June 1, 2009. Online: http://www.vhdl.org/verilog-ams/htmlpages/public-docs/lrm/2.3.1/VAMS-LRM-2-3-1.pdf

[2] Dahlquist, G.; Björck, Å.: Numerical Methods in Scientific Computing:Volume I. SIAM, 2008. Online: http://www.siam.org/books/ot103/

[3] de Boor, C.: A Practical Guide to Splines. Springer, 1978.

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