Subject: Re: Multidimensional Interpolation
Posted by Haje Korth on Tue, 22 Jul 2003 11:51:20 GMT
View Forum Message <> Reply to Message

JD,
could you give me the source for your solution. Are you aware of a good
textbook on this matter?

Thanks,
Haje

"JD Smith" <jdsmith@as.arizona.edu> wrote in message
news:pan.2003.07.21.18.27.19.886736.21942@as.arizona.edu...
> On Mon, 21 Jul 2003 08:40:58 -0700, Ian Chapman wrote:
>
>> Hi,
>>
>> I have created a 5 dimensional data cube (pressure, temperature,
>> relative humidity, frequency, transmission) with a radiative transfer
>> model. I have a user that will need to get transmission data for given
>> values of the rest of the parameters, so I am currently planning to
>> interpolate the cube to the input values of the user.
>>
>> Does anyone know of any multi-dimensional interpolation routines
>> (similar to spline) that would be able to perform this task?
>>
>
>
> You could roll your own using VALUE_LOCATE to locate the point
> (p,t,h,f) within each of the 4 relevant axes (bracketed between
> i,j,k,l and i+1,j+1,k+1,l+1), and then perform quad-linear
> interpolation on the 16 nearby grid points bracketing the desired
> value. E.g., let:
>
> a=(p-p[i])/(p[i+1]-p[i])
> b=(t-t[i])/(t[j+1]-t[j])
> c=(h-h[i])/(h[k+1]-h[k])
> d=(f-f[i])/(f[l+1]-f[l])
>
> The quad-linear interpolant over you whole data cube "z" would look
> like:
>
> (1-a)(1-b)(1-c)(1-d) z[i ,j ,k ,l] +
> (1-a)(1-b)(1-c) d z[i ,j ,k ,l+1] +
> (1-a)(1-b) c (1-d) z[i ,j ,k+1,l] +
> (1-a)(1-b) c d z[i ,j ,k+1,l+1] +
> (1-a) b (1-c)(1-d) z[i ,j+1,k ,l] +

```
> (1-a)   b (1-c)   d  z[i  ,j+1,k  ,l+1] +
> (1-a)   b    c (1-d) z[i  ,j+1,k+1,l  ] +
> (1-a)   b    c    d  z[i  ,j+1,k+1,l+1] +
>    a (1-b)(1-c)(1-d) z[i+1,j  ,k  ,l  ] +
>    a (1-b)(1-c)   d  z[i+1,j  ,k  ,l+1] +
>    a (1-b)   c (1-d) z[i+1,j  ,k+1,l  ] +
>    a (1-b)   c    d  z[i+1,j  ,k+1,l+1] +
>    a    b (1-c)(1-d) z[i+1,j+1,k  ,l  ] +
>    a    b (1-c)   d  z[i+1,j+1,k  ,l+1] +
>    a    b    c (1-d) z[i+1,j+1,k+1,l  ] +
>    a    b    c    d) z[i+1,j+1,k+1,l+1]
>
> The regularity of this pattern lends one to believe a generic n-linear
> interpolation code could be written.  Fancier interpolation methods
> (cubic, spline, sinc) get much harder in higher dimensions.
>
> JD
```