Re: interpolation w/ cubic convolution kernel: boundary treatment?

Source: http://sci.tech−archive.net/Archive/sci.math.num−analysis/2007−04/msg00212.html

- *From*: Markus <iandjohn@xxxxxxxxx>

- *Date*: 26 Apr 2007 15:30:35 −0700

This same topic has been posted just before in another thread.
I am sorry for this, buit it happened due to the very slow response of Google: It took Google 20 hours to display my message after I had posted it.
If you want to reply, please reply to the other thread "cubic convolution interpolation at boundaries?"

Markus

On Apr 26, 2:18 am, Thomas Kluge <thomas.kl...@xxxxxxx> wrote:

> Hi,
>
> I am using the symmetric cubic convolution kernel ("Catmull−Rom splines") to interpolate data over a limited range in a variable x. For the interpolation I am using typically 10 nodes which are equidistant in x.
> Example: The interpolated function between nodes 4 and 5 is computed based on the data points at nodes 3,4,5,6
> A nice property of the Catmull−Rom splines is that I get a continuous 1st derivative everywhere.
>
> My question is now: how should I treat the ranges near the boundary?
> With 10 nodes, how should I interpolate the data between node 9 and 10? So far I am using linear interpolation here − but this is conceptually ugly (and it's not precise, although the latter is not my biggest problem since I want a _nice_ solution).
> I am especially worried that in my current approach the interpolated function has no continuous 1st derivative at node 9.
>
> Is there a solution, in which I could use e.g. a non−symmetric convolution kernel to interpolate between nodes 9 and 10, based on the nodes 8,9,10 or maybe 7,8,9,10 − in a way that I get a continuous 1st derivative everywhere?
> I have not found any discussion about the boundary−treatment

in the literature (and neither on the Google−wide web). In image processing sometimes people mirror the image at the boundaries (i.e. they would introduce a hypothetical 11th node for which the data value is set equal to the data at the 9th node and then interpolate using the data points at nodes 8,9,10,11) − but this would not work in my case.

.