Subject: Questions on the subject of Interpolation
Posted by eyuchen on Thu, 20 Mar 2008 07:09:11 GMT
View Forum Message <> Reply to Message

Hi all,

I was having problems on making 2D interpolations on irregular data points. I read the past posts on this news group, I read the articles in David's website, and I read Ken's sample chapter on this subject.

However, there are still some aspects that I would like to ask about.

1. Is it possible to construct an irregular data interpolation scheme that reduced to bilinear interpolation at the limit of regular grid?

Let me elaborate on this a little bit: The interpolation schemes for regular grids all take the neighboring 4-pts to do the job, whereas that for irregular grids takes only 3. Sometimes, we do have cases where the grid is just slightly distorted and a reasonable 4-pt interpolation *seems* possible.(namely, it is possible to define the "neighboring four points," and one starts to wonder if a 4-pt scheme is superior than the 3-pt scheme.

If 3-pt scheme is superior, then we shouldn't be using 4-pt scheme even in the case of regular grids. If 4-pt scheme is better, there should be a way to do 4-pt irregular interpolation, at least when some conditions are met, right?

2. There is a very good property about 1D linear interpolation, which can be shown as follows:

```
IDL> x=dindgen(5)
IDL> y=[1.2,3.2,4.5,6.1,6.2] ; just some arbitrary irregular spaced
data.
IDL> plot, x, y
IDL> print, interpol(y,x,2.1)
      4.6599998
IDL> print, interpol(x,y,4.6599998)
      2.0999999
```

that is, INTERPOL(x,y,INTERPOL(y,x,?))=? (? is just any number or vector.)
I believe this really lies in the fact that the inverse of linear transformation is still a linear transformation. Therefore, a linear mapping from x to y is the inverse of the linear mapping from y to x.
(you can add a /spline keyword and INTERPOL(x,y,INTERPOL(y,x,?))=? won't be true anymore)

Thus, I can claim that linear interpolation in 1-D is a good scheme (even better than spline) in the sense that I can answer the question "what x should I use in order to get y?" in a relative simple way when I use IDL. (if you use spline, then you'll probably need to solve for a cubic equation to get the job done)

In 2D case, what is the corresponding "good scheme" ? My intuition tells me that the answer is the TRIGRID method, but I would like to discuss with you guys before I feel too certain about it. (Or, shall we suggest ITT built a "backtrace" keyword in all their interpolation routines?)

3. What is the real difference between INTERPOLATE and BILINEAR? It seems to me that INTERPOLATE can do everything BILINEAR does, with more accuracy.

4. What is the real difference between GRIDDATA and TRIANGULATE +TRIGRID? Does the capability of GRIDDATA covers TRIGRID?

5. In GRIDDATA, there is an option "/linear," However, I wonder what does linear means when we only have 3 neighbouring points?

Subject: Re: Questions on the subject of Interpolation
Posted by philip.eisenlohr on Tue, 13 May 2008 21:02:20 GMT
View Forum Message <> Reply to Message

On Mar 20, 9:09 am, eyuc...@gmail.com wrote:
> Hi all,
>
> I was having problems on making 2D interpolations on irregular data
> points. I read the past posts on this news group, I read the articles
> in David's website, and I read Ken's sample chapter on this subject.
>
> However, there are still some aspects that I would like to ask about.
>
> 1. Is it possible to construct an irregular data interpolation scheme
> that reduced to bilinear interpolation at the limit of regular grid?
>
> Let me elaborate on this a little bit: The interpolation schemes for
> regular grids all take the neighboring 4-pts to do the job, whereas
> that for irregular grids takes only 3. Sometimes, we do have cases
> where the grid is just slightly distorted and a reasonable 4-pt
> interpolation *seems* possible.(namely, it is possible to define the
> "neighboring four points," and one starts to wonder if a 4-pt scheme
> is superior than the 3-pt scheme.
>
> If 3-pt scheme is superior, then we shouldn't be using 4-pt scheme

> even in the case of regular grids. If 4-pt scheme is better, there
> should be a way to do 4-pt irregular interpolation, at least when some
> conditions are met, right?
>
> 2. There is a very good property about 1D linear interpolation, which
> can be shown as follows:
>
> IDL> x=dindgen(5)
> IDL> y=[1.2,3.2,4.5,6.1,6.2] ; just some arbitrary irregular spaced
> data.
> IDL> plot, x, y
> IDL> print, interpol(y,x,2.1)
>   4.6599998
> IDL> print, interpol(x,y,4.6599998)
>   2.0999999
>
> that is, INTERPOL(x,y,INTERPOL(y,x,?))=? (? is just any number or
> vector.)
> I believe this really lies in the fact that the inverse of linear
> transformation is still a linear transformation. Therefore, a linear
> mapping from x to y is the inverse of the linear mapping from y to x.
> (you can add a /spline keyword and INTERPOL(x,y,INTERPOL(y,x,?))=?
> won't be true anymore)
>
> Thus, I can claim that linear interpolation in 1-D is a good scheme
> (even better than spline) in the sense that I can answer the question
> "what x should I use in order to get y?" in a relative simple way when
> I use IDL. (if you use spline, then you'll probably need to solve for
> a cubic equation to get the job done)
>
> In 2D case, what is the corresponding "good scheme" ? My intuition
> tells me that the answer is the TRIGRID method, but I would like to
> discuss with you guys before I feel too certain about it. (Or, shall
> we suggest ITT built a "backtrace" keyword in all their interpolation
> routines?)
>
> 3. What is the real difference between INTERPOLATE and BILINEAR? It
> seems to me that INTERPOLATE can do everything BILINEAR does, with
> more accuracy.
>
> 4. What is the real difference between GRIDDATA and TRIANGULATE
> +TRIGRID? Does the capability of GRIDDATA covers TRIGRID?
>
> 5. In GRIDDATA, there is an option "/linear," However, I wonder what
> does linear means when we only have 3 neighbouring points?

I was thinking about the same problem and came to this conclusion:

Imagine a distorted quadrilateral like

```
C------D
|  P  /
|    /
A\  /
  \B
```

where a,b,c,d also serve as actual function values at the nodes with coordinates A_x,A_y and so on.

The bilinearily interpolated value p at position P is given by

p = a + (b-a)x + (c-a)y + (a-b-c+d)xy (see http://en.wikipedia.org/wiki/Bilinear_interpolation)

with [x,y,xy]^T = M * [P_x-A_x,P_y-A_y,(P_x-A_x)(P_y-A_y)]^T (this is my "magic")

and M =

```
[1 0 1]   [    B_x-A_x            C_x-A_x           D_x-
A_x       ]^-1
[0 1 1] * [    B_y-A_y            C_y-A_y           D_y-A_y       ]
[0 0 1]   [(B_x-A_x)(B_y-A_y) (C_x-A_x)(C_y-A_y) (D_x-A_x)(D_y-A_y)]
```

M maps the distorted qudrilateral to a nice unit square which is then connected to the alternative bilinear interpolation function found at http://en.wikipedia.org/wiki/Bilinear_interpolation.

Hope it helps!

Cheers
Philip