
Subject: interpolate weirdness

Posted by [rogass](#) on Fri, 21 Feb 2014 12:00:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi Folks,

I know that there is no 'correct' way for interpolation, but maybe a 'best' IDL way. The following example easily demonstrates what I mean:

```
IDL> d=dist(20) & print,interpolate(d,[.5],[.5],/grid)
0.853553
```

```
IDL> print,interpolate(d,[.5],[.5],/grid,/cubic)
0.622236
```

```
IDL> print,interpolate(d,[.400544],[.400544],/grid)
0.707107
```

```
IDL> print,interpolate(d,[.581946],[.581946],/grid,/cubic)
0.707107
```

```
IDL> print,'The result should be: ',sqrt(.5)
The result should be:    0.707107
```

It does not play a role which IDL version is used - for IDL 6.4 and for IDL 8.3 I got the same results. Is there any way to get 'closer' to a reasonable result?

Thanks and Cheers

Chris

Subject: Re: interpolate weirdness

Posted by [Helder Marchetto](#) on Fri, 21 Feb 2014 12:33:45 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Friday, February 21, 2014 1:00:36 PM UTC+1, CR wrote:

> Hi Folks,

>

>

>

> I know that there is no 'correct' way for interpolation, but maybe a 'best' IDL way. The following example easily demonstrates what I mean:

>

>

>

> IDL> d=dist(20) & print,interpolate(d,[.5],[.5],/grid)

```

>
> 0.853553
>
>
>
> IDL> print,interpolate(d,[.5],[.5],/grid,/cubic)
>
> 0.622236
>
>
>
> IDL> print,interpolate(d,[.400544],[.400544],/grid)
>
> 0.707107
>
>
>
> IDL> print,interpolate(d,[.581946],[.581946],/grid,/cubic)
>
> 0.707107
>
>
>
> IDL> print,'The result should be: ',sqrt(.5)
>
> The result should be: 0.707107
>
>
>
>
>
> It does not play a role which IDL version is used - for IDL 6.4 and for IDL 8.3 I got the same
> results. Is there any way to get 'closer' to a reasonable result?
>
>
>
> Thanks and Cheers
>
>
>
> Chris

```

I'm not sure what you're trying to do here, but interpolating near borders is kind of complicated because you can only have one side to rely on. However, if you're going to use bicubic interpolation, I would recommend using:

```
print,interpolate(d,[.5],[.5],/grid,cubic=-0.5)
```

then you get:

```
0.738791
```

Hope it helps,
h

Subject: Re: interpolate weirdness
Posted by [Helder Marchetto](#) on Fri, 21 Feb 2014 13:59:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Friday, February 21, 2014 1:33:45 PM UTC+1, Helder wrote:

> On Friday, February 21, 2014 1:00:36 PM UTC+1, CR wrote:

>

>> Hi Folks,

>

>>

>

>>

>

>>

>

>> I know that there is no 'correct' way for interpolation, but maybe a 'best' IDL way. The following example easily demonstrates what I mean:

>

>>

>

>>

>

>>

>

>> IDL> d=dist(20) & print,interpolate(d,[.5],[.5],/grid)

>

>>

>

>> 0.853553

>

>>

>

>>

>

>>

>

>> IDL> print,interpolate(d,[.5],[.5],/grid,/cubic)

>

>>

>

>> 0.622236

>

>>

>

```

>>
>
>>
>
>> IDL> print,interpolate(d,[.400544],[.400544],/grid)
>
>>
>
>> 0.707107
>
>>
>
>>
>
>>
>
>> IDL> print,interpolate(d,[.581946],[.581946],/grid,/cubic)
>
>>
>
>> 0.707107
>
>>
>
>>
>
>> IDL> print,'The result should be: ',sqrt(.5)
>
>>
>
>> The result should be: 0.707107
>
>>
>
>>
>
>>
>
>>
>
>>
>
>> It does not play a role which IDL version is used - for IDL 6.4 and for IDL 8.3 I got the same
results. Is there any way to get 'closer' to a reasonable result?
>
>>

```

>
>>
>
>>
>
>> Thanks and Cheers
>
>>
>
>>
>
>>
>
>> Chris
>
>
>
> I'm not sure what you're trying to do here, but interpolating near borders is kind of complicated because you can only have one side to rely on. However, if you're going to use bicubic interpolation, I would recommend using:
>
> print,interpolate(d,[.5],[.5],/grid,cubic=-0.5)
>
> then you get:
>
> 0.738791
>
> Hope it helps,
>
> h

Just for sake of comparison, if you do the same for a point not next to the boundary, you get:
print,interpolate(d,[5.5],[5.5],/grid, cubic=-0.5)

7.77795

print, sqrt(2)*5.5

7.77817

The difference is 0.00022220612, what is much better.

Hope it helps.

Subject: Re: interpolate weirdness

Posted by [Helder Marchetto](#) on Fri, 21 Feb 2014 14:06:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Friday, February 21, 2014 2:59:46 PM UTC+1, Helder wrote:

> On Friday, February 21, 2014 1:33:45 PM UTC+1, Helder wrote:

>

```
>> On Friday, February 21, 2014 1:00:36 PM UTC+1, CR wrote:
>
>>
>
>>> Hi Folks,
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> I know that there is no 'correct' way for interpolation, but maybe a 'best' IDL way. The
following example easily demonstrates what I mean:
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> IDL> d=dist(20) & print,interpolate(d,[.5],[.5],/grid)
>
>>
>
>>>
>
>>
>
>>> 0.853553
>
>>
```

```
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> IDL> print,interpolate(d,[.5],[.5],/grid,/cubic)
>
>>
>
>>>
>
>>
>
>>> 0.622236
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> IDL> print,interpolate(d,[.400544],[.400544],/grid)
>
>>
>
>>>
>
>>
>
>>> 0.707107
>
>>
```

```
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> IDL> print,interpolate(d,[.581946],[.581946],/grid,/cubic)
>
>>
>
>>>
>
>>
>
>>> 0.707107
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> IDL> print,'The result should be: ',sqrt(.5)
>
>>
>
>>>
>
>>
>
>>> The result should be: 0.707107
>
>>
```

```
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>> It does not play a role which IDL version is used - for IDL 6.4 and for IDL 8.3 I got the same
results. Is there any way to get 'closer' to a reasonable result?
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> Thanks and Cheers
>
>>
>
>>>
>
>>
>
>>>
>
```

```

>>
>
>>>
>
>>
>
>>> Chris
>
>>
>
>>
>
>>
>
>> I'm not sure what you're trying to do here, but interpolating near borders is kind of complicated
because you can only have one side to rely on. However, if you're going to use bicubic
interpolation, I would recommend using:
>
>>
>
>> print,interpolate(d,[.5],[.5],/grid,cubic=-0.5)
>
>>
>
>> then you get:
>
>>
>
>> 0.738791
>
>>
>
>> Hope it helps,
>
>>
>
>> h
>
>
>
> Just for sake of comparison, if you do the same for a point not next to the boundary, you get:
>
> print,interpolate(d,[5.5],[5.5],/grid, cubic=-0.5)
>
> 7.77795
>
> print, sqrt(2)*5.5
>

```

> 7.77817
>
> The difference is 0.00022220612, what is much better.
>
>
>
> Hope it helps.

One more comment about the bilinear interpolation:

```
IDL> d=dist(20) & print,interpolate(d,[.5],[.5],/grid)
0.853553
```

The value 0.853553 is exactly what is expected!

Linear interpolation on x gives 0.5 and $(1+\sqrt{2})/2=2.414$, then take the average of those two values and you get:

```
IDL> (0.5+(1.0+sqrt(2))/2.0)/2.0
0.85355341
```

Correct answer. See http://en.wikipedia.org/wiki/Bilinear_interpolation

Cheers,
Helder

Subject: Re: interpolate weirdness
Posted by [rogass](#) on Fri, 28 Feb 2014 16:31:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear Helder,
thank you for clarification. The situation has changed. Desired values are now far away from boundaries and I can use something like this:

```
function cr_interpol,dat,mat_old,mat_new,cubic=cubic

ndim = size(dat,/ndimensions)
dims = size(dat,/dimensions)
case (1b) of
ndim eq 1 : begin
    newx = (interpol(ulindgen(dims[0]),(mat_old[0,*])[*],(mat_new[0,*]) [*]))[*]
    result = interpolate(dat[*],newx,/grid,cubic=cubic)
end
ndim eq 2 : begin
    newx = (interpol(ulindgen(dims[0]),(mat_old[0,*])[*],(mat_new[0,*]) [*]))[*]
    newy = (interpol(ulindgen(dims[1]),(mat_old[1,*])[*],(mat_new[1,*]) [*]))[*]
    result = interpolate(dat,newx,newy,/grid,cubic=cubic)
end
ndim eq 3 : begin
    newx = (interpol(ulindgen(dims[0]),(mat_old[0,*])[*],(mat_new[0,*]) [*]))[*]
    newy = (interpol(ulindgen(dims[1]),(mat_old[1,*])[*],(mat_new[1,*]) [*]))[*]
```

```
    newz = (interpol(ulindgen(dims[2]),(mat_old[2,*])[*],(mat_new[2,*]) [*]))[*]  
    result = interpolate(dat,newx,newy,newz,/grid,cubic=cubic)  
end  
endcase  
return,result  
end
```

Cheers

Chris
