
Subject: Re: Reverse interpolation?

Posted by [Craig Markwardt](#) on Wed, 16 Aug 2000 07:00:00 GMT

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

Martin Schultz <martin.schultz@dkrz.de> writes:

> Simon de Vet wrote:

...

>> I want to get better results. Instead of finding that entry #5 is the
>> closest match to my known altitude, and using #5 as the subscript in my
>> data array, I'd like to find that, based on a linear interpolation,
>> entry #5.32850 is an exact match, and using this value in an
>> interpolation on my data array.

>>

>> I understand how to use a subscript to find an interpolated value. How
>> would I go in the opposite direction, using a value to find a subscript?

>>

>> Thanks. I hope this isn't too confusing. I'm having a bad brane day.

>>

>> Simon

>

>

> Hi Simon,

>

> I have dealt with similar problems (vertical regridding, to
> give the kid a name), and I tend to resort on the spline routine
> in these instances. Please find attached a piece of code that I
> used to vertically regrid OH concentration fields. Just be
> careful about end effects. For this application I overwrote the
> topmostfour levels of the target grid with the values from the
> top level of the old grid.

>

...

> newxx = Spline(oldy, oldx, newy, 5.)

...

If you are trying to match a lot of values, then I agree that the spline interpolation is one of the best solutions. However, I have had bad experiences with the SPLINE function. It tends to have problems when the points are too far apart, and it's also pretty slow. I found the Numerical Recipes version of spline interpolation to be much better (built in functions SPL_INTERP and SPL_INIT).

If v1 is vector of values sampled at original altitudes a1, and you want to resample at new altitudes a2, then you do something like this:

```
v2 = spl_interp(a1, v1, spl_init(a1, v1), a2)
```

Now v2 has the newly sampled values. In principle a2 can be one value so even if you want a single value the spline interpolation can be useful, and even pretty fast.

IDL version 5.3 apparently has a new function called VALUE Locate which can do fast search of a reference array. From there you can do a linear interpolation. The IDL Astronomy Library procedure called TABINV may do what you want. The versions that use VALUE Locate and don't use it are here, and here, respectively:

<http://idlastro.gsfc.nasa.gov/ftp/v53/tabinv.pro>
<http://idlastro.gsfc.nasa.gov/ftp/pro/math/tabinv.pro>

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

Subject: Re: Reverse interpolation?
Posted by [promashkin](#) on Wed, 16 Aug 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

I'd recommend checking out VALUE Locate function. Overall, solving the proportion in a triangle will be the simplest approach (unless you want to do a more sophisticated spline as suggested by Martin), but VALUE Locate eliminates the need for subtracting and finding near-zero elements of the array.

Also, once you have X, there is a library function INTERPOL that will take X for input and locate the interpolated value for you.

Cheers,
Pavel

Simon de Vet wrote:

>

> Simon de Vet wrote:

>

>> I understand how to use a subscript to find an interpolated value. How
>> would I go in the opposite direction, using a value to find a subscript?

>

> I have an idea (I used this for a similar problem with Matlab) that may
> work. It's awkward and bulky, but I can understand it.

>

> First, I subtract the known altitude from the altitude list. Some of the
> values will be positive, and some negative. I use this to find the points

> on either side of known altitude. I can use their subscripts as x-values
> and their altitudes as y-values to find the equation of a line connecting
> the two points. I can then substitute my known altitude (y value) into the
> equation, and solve for x (the required subscript).
>
> I hope it works.
>
> Simon

Subject: Re: Reverse interpolation?
Posted by [Simon de Vet](#) on Wed, 16 Aug 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Simon de Vet wrote:

> I understand how to use a subscript to find an interpolated value. How
> would I go in the opposite direction, using a value to find a subscript?

I have an idea (I used this for a similar problem with Matlab) that may work. It's awkward and bulky, but I can understand it.

First, I subtract the known altitude from the altitude list. Some of the values will be positive, and some negative. I use this to find the points on either side of known altitude. I can use their subscripts as x-values and their altitudes as y-values to find the equation of a line connecting the two points. I can then substitute my known altitude (y value) into the equation, and solve for x (the required subscript).

I hope it works.

Simon

Subject: Re: Reverse interpolation?
Posted by [Martin Schultz](#) on Wed, 16 Aug 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Simon de Vet wrote:

>
> I have a little problem :)
>
> I have three sets of data - a list of altitudes (which are not evenly
> spaced, and are calculated with a long, confusing formula), an altitude
> I want to match, and a set of data that depends on altitude (ie: entry 1
> corresponds to altitude 1, entry 2 to altitude 2, etc...).

>
 > Currently, I am calculating the differences between each entry in the
 > altitude list and my known altitude, and using this to find the closest
 > match, who's subscript I use in the data array. However, since there
 > aren't very many values in my altitude list, the closest matches are
 > often not very close at all, and the results become inaccurate.
 >
 > I want to get better results. Instead of finding that entry #5 is the
 > closest match to my known altitude, and using #5 as the subscript in my
 > data array, I'd like to find that, based on a linear interpolation,
 > entry #5.32850 is an exact match, and using this value in an
 > interpolation on my data array.
 >
 > I understand how to use a subscript to find an interpolated value. How
 > would I go in the opposite direction, using a value to find a subscript?
 >
 > Thanks. I hope this isn't too confusing. I'm having a bad brane day.
 >
 > Simon

Hi Simon,

I have dealt with similar problems (vertical regridding, to give the kid a name), and I tend to resort on the spline routine in these instances. Please find attached a piece of code that I used to vertically regrid OH concentration fields. Just be careful about end effects. For this application I overwrote the topmostfour levels of the target grid with the values from the top level of the old grid.

Cheers,
 Martin

```
FOR i=0,N_Elements(grid.lon)-1 DO BEGIN
  FOR j=0,N_Elements(grid.lat)-1 DO BEGIN

    oldx = reverse( reform( newoh[i,j,*,month-1] ) )
    oldy = reverse( reform( data.p ) )
    newy = reform( pgrid[i,j,*,month-1] )

    ;; Smooth spline interpolation onto new grid
    newxx = Spline( oldy, oldx, newy, 5. )

    ;; Enter new values in result field
    resultoh[i,j,*,month-1] =
```

--

✓

Sorry, I've never needed any extra smoothing, and I have always had control over the endpoints of my original function.

—

Craig Markwardt wrote:

```
>
> Martin Schultz <martin.schultz@dkrz.de> writes:
>
> Craig wrote:
>
>>
> ...
>>      newxx = Spline( oldy, oldx, newy, 5. )
> ...
>
> If you are trying to match a lot of values, then I agree that the
> spline interpolation is one of the best solutions. However, I have
> had bad experiences with the SPLINE function. It tends to have
> problems when the points are too far apart, and it's also pretty slow.
> I found the Numerical Recipes version of spline interpolation to be
> much better (built in functions SPL_INTERP and SPL_INIT).
>
> If v1 is vector of values sampled at original altitudes a1, and you
> want to resample at new altitudes a2, then you do something like this:
>
> v2 = spl_interp(a1, v1, spl_init(a1, v1), a2)
>
> Now v2 has the newly sampled values. In principle a2 can be one value
> so even if you want a single value the spline interpolation can be
> useful, and even pretty fast.
>
```

I had looked into SPL_INTERP and SPL_INIT, but I found the problem was that it does not allow for "smoothing" (more correctly: controlling "the amount of tension that is applied to the curve"). I certainly like the idea of controlling the endpoints (which you can do only in SPL_INTERP). Are you aware of a solution that can do both?

Martin

--

```
[[[
[[ Dr. Martin Schultz  Max-Planck-Institut fuer Meteorologie
[[
[[      Bundesstr. 55, 20146 Hamburg
[[
[[      phone: +49 40 41173-308
[[
[[      fax:  +49 40 41173-298
[[
```

[[

[illegible]