Subject: Re: Don't Extrapolate with GRID3 (or cut off excess)
Posted by asdf on Mon, 02 Aug 2010 14:50:51 GMT
View Forum Message <> Reply to Message

On Jul 30, 11:48 am, Paulo Penteado <pp.pente...@gmail.com> wrote:
> On Jul 30, 12:33 pm, asdf <rmc...@gmail.com> wrote:
>
>> First, a little background. I have an irregular grid of 3D volume data
>> that I would like to visualize (currently using iVolume). The volume
>> is basically a wedge piece of a cylinder.
>> I've the placed data onto a regular, cube grid using GRID3, but the
>> result has a large, unrealistic value in an area outside of the
>> original irregular grid. From GRID3 documentation, it doesn't seem
>> like I can force it to ignore points outside the original grid, i.e.
>> don't extrapolate (correct?). I'm wondering if someone knows an
>> efficient way to set to NaN values in the regular grid that are
>> outside the original, irregular grid. It seems I can do a Delaunay
>> triangulation of the irregular grid, and use it somehow to find if a
>> point in the regular grid is inside the irregular grid, but I'm lost
>> on the details of how to do that.
>
> If your distribution is convex, you can get what you want with
> qgrid3(), instead of grid3(), as it allows to set a value for the
> points outside the convex hull.
>
> If it is not convex, things are trickier. If the original points all
> fall into a series of planes, you could make an IDLanROIGroup with
> those, then use the containspoints method to determine which points of
> the grid fall inside it.

Thanks for the reply, that works great:

```
qhull,x,y,z,tr,/delaunay
result=qgrid3(x,y,z,data,tr,missing=!values.d_nan)
```

Even faster than grid3 also.