Subject: Re: XYZ plot + Normal to surface output
Posted by Rick Towler on Mon, 07 Jan 2008 23:19:49 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> a.lucas writes:

>
>> 1/ I have some xyz ASCII files and I want to plot them in a 3D view.
>> But my X and Y are not regular. Is there any way to plot them firstly
>> without any interpolation and secondly using kriging interpolation so
>> as to get regular plot.
>
> Irregular data is not a problem. But I would use object
> graphics (not necessarily a subject for beginners) to do the plotting.
> You could try iSurface, or (if you wanted something you could actually
> learn from) you could try FSC_SURFACE:
>
> http://www.dfanning.com/programs/fsc_surface.zip

Yes, object graphics is definitely the way to go.

>> 2/ I have two files (also in XYZ ASCI format), corresponding to Vx
>> and Vy of a velocity field. How can I plot the complet velocity field
>> with vectors in 2D and draped on a 3D surface ?
>
> Well, this is an ADVANCED topic, for sure. I'm not sure how I would
> do this. Maybe later I'll have a chance to think about it more.
> STREAMLINE comes to mind, and it is easy enough to draw lines
> in 3D, but something that looks good... I don't know. Maybe
> the more advanced users in the group will have some ideas. :-)

You could do this one of two ways.

The easy way would be to generate a texture map of your vectors and apply that to your IDLgrSurface object. You'll want to look at the TEXTURE_HIRES keyword and consider your hardware capabilities when going this route. It may work, or it may look terrible. But it is easy.

The other way would be to create a vector field object that is a subclass of IDLgrModel that accepts your vector data, as well as the IDLgrSurface object as inputs and for each vector it extracts the Z data for the head and tail of the vector and draws a vector at each grid point. You would of course have to extract the surface Z value at the vector tail and all of the neighboring points, determine the two neighboring points the head falls between, calculate the normal of the polygon bound by the tail point and these points, calculate the Z value

of the head based on the magnitude of the vector, draw the vector with the head orthogonal to your normal, then shift the vector up in Z just a smidgen so it floats just above your surface. This would give you the best looking results but takes more effort up front. It would be *much* easier if you were working with data on the same grids but I don't think it is required.

-Rick
