Subject: Adaptation of surf_track for viewing earthquakes in 3 dimensions
Posted by twright on Thu, 11 Jul 2002 08:02:37 GMT
View Forum Message <> Reply to Message

I have been developing the means to view and run animations of earthuake datasets under a topographic surface. One driver program calls Surf_track to display single or multiple earthquake datasets that can be freely rotated with the mouse. Animations are run using a driver program that calls surf_track in a loop that converts the 3-d object view to a 2-d view, then loads an mpeg file run by xinteranimate. Several questions have come up in the course of this development.

1.IDL seems peculiarly unsuited for use with 3-dimensional point data. I have no need of connecting earthquakes, contouring surfaces, or otherwise looking at them other than as points in x-y-z space. I had expected to be able to freely configure symbol size, symbol shape and color to represent different earthquake parameters such as magnitude, depth, time intervals or other user-defined classification parameters. But none of the IDLgr routines allows this. IDLgrpolyline allows symbol manipulation, but only one color. IDLgrpolygon accepts a color matrix using vertex_colors, but has no call to symbol. Idlgrplot has both symbol and color control, but the z parameter, if used, is set to a constant.

My question: has anyone developed an idl routine that has full capacity to deal with 3-d point data? If not, would any idl developer be willing to modify one of the above routines to satsify this need? For instance, modify idlgrplot to accept x-y-z data as input?

2. A second question is aimed at "DD", the developer of surf_track. First, my congratulations on writing a very complex and very useful program. There is a rotation matrix in the program used to specify the starting point of view before free rotation using the mouse.
ogroup->rotate,[1,0,0],0
ogroup->rotate,[0,1,0],0
ogroup->rotate,[1,0,0],0
For example, this yields a view looking vertically down.

It would be very useful to have the rotation parameters written on the screen (or included in one of the "other options" buttons) while rotating the image using the mouse. Then, for animation purposes, one could find an optimum viewing angle for the 2-d animation and load the three parameters into surf_track without a lot of guesswork and trial and error. I would be happy to implement this if given the code, but I am not yet proficient enough to figure this one out.

Thanks for any information on either of these topics.