Subject: Re: 3D plot of set of curves
Posted by David Fanning on Mon, 24 May 2004 23:33:34 GMT
View Forum Message <> Reply to Message

Timm Weitkamp writes:

> I have a set of curves -- say, the values are in a 2D array "Z" -- and
> would like to make a nice-looking graph something like this hand-drawn
> sketch:
>
> http://tinyurl.com/36frg
>
> (NB: On Sat-Sun 29-30 May it's probably useless trying this link -- they
> tell me our web server will be down. Sorry about that.)
>
> In what way exactly the "walls" are shaded is not crucial, but I
> definitely want the "base lines" (i.e., the lines at z=0) drawn, with
> hidden lines removed, as in the sketch figure referenced above.
>
> I looked into the help for SURFACE and SHADE_SURF, but did not find what I
> need. The HORIZONTAL keyword to SURFACE goes in the right direction, but
> does only does a disappointingly small part of the job.
>
> Is there a not-too-cumbersome way in IDL to do what I want?

I'm not too sure what "not-too-cumbersome" means in this
context. Is there a built-in IDL routine to do it? I don't
think so. I've never seen one. Can I imagine writing such
a routine? Yes. I'd definitely do it in object graphics,
because the 3D part of it will be much easier, and you
will be able to rotate it, which will make it easier for
the user to see different parts of it.

It will involve some interesting programming. Some people
consider that cumbersome, but no one on this newsgroup. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: 3D plot of set of curves

On 24.05.04 at 17:33 -0600, David Fanning wrote:

> Timm Weitkamp writes:
>
>> I have a set of curves -- say, the values are in a 2D array "Z" -- and
>> would like to make a nice-looking graph something like this hand-drawn
>> sketch:
>>
>> http://tinyurl.com/36frg
>>
>> [...]

> Is there a built-in IDL routine to do it? I don't
> think so. I've never seen one. Can I imagine writing such
> a routine? Yes. I'd definitely do it in object graphics,
> because the 3D part of it will be much easier, and you
> will be able to rotate it, which will make it easier for
> the user to see different parts of it.

It finally turned out not to be so hard, yet without any object graphics.
The T3D mechanism provides enough rotation functionality for my purposes.

Below is the code I am now using. Poorly documented, unreliable, no
_EXTRAs, etc. But it does what I want. And there is an example at the end,
for whoever may want to look at it :-)

Timm

Timm Weitkamp <http://people.web.psi.ch/weitkamp>

```
;; ---------------------- Start ----------------------------

PRO PlotCurveSet, z, x, y $
                , FILL_COLORS = fillColors $
                , VERTICAL_BARS = vertBars $
                , PSYM = psym

dimZ = SIZE(z, /DIMENSIONS)
nx = dimZ[0]
ny = dimZ[1]

IF N_ELEMENTS(x) EQ 0 THEN x = FINDGEN(nx)
IF N_ELEMENTS(y) EQ 0 THEN y = FINDGEN(ny)
IF N_ELEMENTS(fillColors) EQ 0 THEN fillColors = LONARR(nx)
```

```
;;  Set up coordinate axes

SURFACE, /NODATA, /SAVE, z, x, y


;;  Draw "sheets" from back to front

FOR ix = nx-1, 0, -1 DO BEGIN

   polyY = [y, REVERSE(y), y[0]]
   polyX = x[ix] + FLTARR(N_ELEMENTS(polyY))
   polyZ = [REFORM(z[ix,*]), FLTARR(ny), z[ix,0]]

   POLYFILL, polyX, polyY, polyZ, /T3D, COLOR = fillColors[ix]
   PLOTS, polyX, polyY, polyZ, /T3D, PSYM=psym

   IF KEYWORD_SET(vertBars) THEN FOR iy = 0, ny-1 DO $
     PLOTS, x[[ix,ix]], y[[iy, iy]], [0, z[ix,iy]], /T3D

ENDFOR

END


;;  ---------------------- Example --------------------------

z = (SIN((FINDGEN(25)/75) ## [1,2,3,4,5,6,7,8] ))^2
PlotCurveSet, z, /VERTICAL_BARS

END

;; ------------------------ End ---------------------------
```