
Subject: arrow.pro to 3-D

Posted by [Nic](#) on Sat, 19 Aug 2006 00:16:43 GMT

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

Hi again! I have been trying to convert arrow.pro to 3D. Here's what I have so far. When I try it out, it compiles. I tested it out and my output sort of makes sense.

Usage:

Arrow_3d x0, y0, z0, x1, y1, z1

If I input:

```
> arrow_3d, 100, 200, 0, 200, 200, 0      I get arrow to the right. check
> arrow_3d, 100, 200, 0, 100, 300, 0      I get arrow up. check
> arrow_3d, 100, 200, 0, 100, 200, 300    I get a dot.
```

Is the dot really an arrow pointing out of the screen in a 2-d plots, or have I screwed up my modification?

Thank you,
Nic

```
PRO ARROW_3D, x0, y0, z0, x1, y1, z1, HSIZE = hsize, COLOR = color,
HTHICK = hthick, $
  THICK = thick, DATA = data, NORMALIZED = norm, $
  SOLID = solid
```

```
COMPILE_OPT idl2
```

```
ON_ERROR, 2
; Set up keyword params
```

```
if n_elements(thick) eq 0 then thick = 1.
if n_elements(hthick) eq 0 then hthick = thick
```

```
  ;Head size in device units
if n_elements(hsize) eq 0 then arrowsize = !d.x_size/64. * (hthick/2. >
1) $
  else arrowsize = float(hsize)
if n_elements(color) eq 0 then color = !P.color
mcost = -.866d ;We use 30 degrees for head angle
sint = .500d
msint = - sint
```

```
for i = 0L, n_elements(x0)-1 do begin ;Each vector
```

```

if keyword_set(data) then $ ;Convert?
  p = convert_coord([x0[i],x1[i]], [y0[i],y1[i],[z0[i],z1[i]]],
/data,/t3d, /to_dev) $
else if keyword_set(norm) then $
  p = convert_coord([x0[i],x1[i]], [y0[i],y1[i],[z0[i],z1[i]] /norm,
/to_dev) $
else p = [[x0[i], y0[i], z0[i]], [x1[i], y1[i], z1[i]]]

xp0 = p[0,0]
xp1 = p[0,1]
yp0 = p[1,0]
yp1 = p[1,1]
zp0 = p[2,0]
zp1 = p[2,1]

dx = xp1 - xp0
dy = yp1 - yp0
dz = zp1 - zp0
sep = sqrt(dx^2d + dy^2d + dz^2d) ;Length

if sep gt 0 then begin
  dx = dx/sep ;Cos th
  dy = dy/sep ;Sin th
  dz = dz/sep ;Sin th (?) check
endif else begin
  dx = 1.
  dy = 0.
  dz = 0.
  sep = 1.
endelse
if arrowsize gt 0 then a = arrowsize $ ;a = length of head
else a = -sep * arrowsize

xxp0 = xp1 + a * (dx*mcost - dy * msint)
yyp0 = yp1 + a * (dx*msint + dy * mcost)
xxp1 = xp1 + a * (dx*mcost - dy * sint)
yyp1 = yp1 + a * (dx*sint + dy * mcost)

if keyword_set(solid) then begin ;Use polyfill?
  b = a * mcost*.9d ;End of arrow shaft (Fudge to force join)
  plots, [xp0, xp1+b*dx], [yp0, yp1+b*dy],[zp0, zp1+b*dz] /DEVICE, $
  COLOR = color, THICK = thick
  polyfill, [xxp0, xxp1, xp1, xxp0], [yyp0, yyp1, yp1, yyp0] $
  /DEVICE, COLOR = color
endif else begin
  plots, [xp0, xp1], [yp0, yp1], [zp0, zp1], COLOR = color, THICK =
thick, /DEVICE
  plots, [xxp0,xxp1,xxp1],[yyp0,yp1,yyp1], /DEVICE, COLOR = color, $

```

```
    THICK = hthick
endelse
ENDFOR
end
```

Norbert Hahn wrote:

> "Nic" <nicole_messages@juno.com> wrote:

>

>> It is making lines (how do I get it to plot arrows?), but I think each

>> time the for loop runs, it is overwriting the previous loop's plots. I

>> want to overplot each new plots to the original surface plot.

>

> You may use arrow in stead of plots. arrow can be called as often as needed

> and will add one or more arrows to an existing plot. Unfortunately the

> coordinates accepted by arrow are 2D. So you need either transform your

> 3D data to 2D by calling convert_coord or modify the call to convert_coord

> within arrow.pro.

>

> HTH

> Norbert