
Subject: Re: mesh_volume and tetra_volume
Posted by [Karl Schultz](#) on Wed, 18 Aug 2004 17:34:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

In your specific case, if you have a volume that describes an object in such a way that the volume is filled with 0's where the object does not exist, and then 1's where it does exist, then there is no "radius". You should be able to just use the isovalue of 1. This will place the isosurface on the outer surface of your "object", which is what I think you want.

Example:

```
IDL> vol = bytarr(30,30,30)
IDL> vol[13:15, 13:15, 13:15] = 1
IDL> isosurface, vol, 1, v, c
IDL> xobjview, OBJ_NEW('IDLgrPolygon', v, polygons=c)
IDL> print, MESH_VOLUME(v,c)
0.000000
IDL> r = MESH_VALIDATE(v,c,/combine)
IDL> print, MESH_VOLUME(v,c)
8.00001
```

I set 3x3x3 samples near the middle of the volume to 1 to represent the extents of my little cube. The extents enclose a volume of 2x2x2, which is 8.

Here's some more info that I thought I'd pass along.

People generate isosurfaces to look at volume data collected from some sort of device. And they know precisely what the values in volume represent. For example, if you have a 3D scan of a human head, you might know that skin values are represented in the range 10-20 (out of a range from 0-255). If you construct an isosurface from this volume using an isovalue of 18 or something, you'll get a pretty good surface that approximates the skin. (I'm making up these values)

```
IDL> head = READ_BINARY(FILEPATH('head.dat', SUBDIRECTORY=['examples',
'data']), DATA_DIMS=[80,100,57])
IDL> isosurface, head, 18, v, c
IDL> xobjview, OBJ_NEW('IDLgrPolygon', v, polygons=c)
```

Maybe it is easier to look at a simpler example:

```
IDL> vol = BYTARR(3,3,3)
IDL> vol[1:2,*,*]=1
IDL> isosurface, vol, 0.5, v, c
IDL> print, v
```

```

0.500000 0.000000 0.500000
0.500000 0.500000 0.000000
0.500000 0.000000 0.000000
0.500000 0.000000 1.000000
0.500000 0.500000 1.000000
0.500000 1.000000 0.500000
0.500000 1.000000 0.000000
0.500000 1.000000 1.000000
0.500000 1.500000 0.000000
0.500000 2.000000 0.500000
0.500000 1.500000 1.000000
0.500000 2.000000 1.000000
0.500000 2.000000 0.000000
0.500000 0.000000 1.500000
0.500000 1.000000 1.500000
0.500000 0.500000 2.000000
0.500000 1.000000 2.000000
0.500000 0.000000 2.000000
0.500000 1.500000 2.000000
0.500000 2.000000 1.500000
0.500000 2.000000 2.000000
IDL> xobjview, OBJ_NEW('IDLgrPolygon', v, polygons=c)

```

The volume is a small cube where 1/3 of the volume is a "slab" with sample values of 0, and 2/3 a thicker slab with sample values of 1. If you want to find the isosurface with isovalue of 0.5, it would split the thinner slab in two, because that one-voxel-wide slab would have samples of 0 on one side and samples of 1 on the other. The isosurface, with a value of 0.5, must pass through the volume so that all points on the isosurface are the same distance from samples of value 0 and samples of value 1. That is the basic nature of an isosurface.

So, the isosurface is a simple plane that is located at $x = 0.5$. See the vertex list above. You can view it with `xobjview`. You'll need to rotate it because it is edge-on and set the drag quality to low if you want to see the mesh.

It just happens that the sample values in this case coincide with the volume coords. I could also do:

```

IDL> vol = bytarr(3,3,3)
IDL> vol[0,*,*] = 200
IDL> vol[1:2,*,*]=201
IDL> isosurface, vol, 200.5, v, c
IDL> print,v

```

and get the exact same vertices as above. The point is that the isovalue is exactly the average of the samples at `vol[0,*,*]` and `vol[1,*,*]` and so the

isosurface must pass through [0.5, *, *].

Now let's try an interval volume.

```
IDL> vol = BYTARR(3,3,3)
IDL> vol[1,*,*]=1
IDL> vol[2,*,*]=2
IDL> interval_volume, vol, 0.5, 1.5, v, c
IDL> c2 = tetra_surface(v,c)
IDL> xobjview, OBJ_NEW('IDLgrPolygon', v, polygons=c2)
```

Same idea here except that my volume is three slabs with values 0, 1, and 2. I constructed an interval volume that bisects each slab along the X direction. So I end up with a box that extends from 0.5 to 1.5 in X and covers the entire volume in Y and Z.

I hate to repeat myself, but you really have to know your data and know what you want to get out of it if you intend to make isosurface and interval volume work for you. Outside of some test volumes we've discussed, I'm not clear on what you are trying to do.

If we go back to the sphere:

```
n = 40
mid = 20
radius = 10
vol = BYTARR(n,n,n)
for i=0, n-1 do begin
  for j=0, n-1 do begin
    for k=0, n-1 do begin
      vol[i,j,k] = SQRT( (i-mid)^2 + (j-mid)^2 + (k-mid)^2 ) + 0.5
    endfor
  endfor
endfor
```

```
ISOSURFACE, vol, radius, v, c
```

```
print, "Ideal volume", 4./3.*!pi*radius^3
print, 'Isosurface is ', MESH_ISSOLID(c) ? 'solid' : 'not solid'
print, "Isosurface volume", MESH_VOLUME(v, c)
```

```
Ideal volume 4188.79
Isosurface is not solid
Isosurface volume 0.000000
```

So, my volume is 40x40x40 and I fill in the sample values so that each sample value is the distance from the "center" that I have chosen at 20,20,20. So the volume is now a "field" of samples where the sample value

is zero at 20,20,20 and the sample values increase as they are farther from the center. The maximum value is about 35 ($20 * \sqrt{3}$) and occurs in the corners of the volume.

Now I can pick any isovalue I want. If I pick something bigger than 35, there are no samples in the volume larger than 35, so ISOSURFACE will not generate a surface. For example, if I pick 45, the isovalue of 45 does not fall between any two samples in the volume. If I pick something between 20 and 35, I get a clipped sphere.

But if I pick something like 10, I should get a solid sphere. But I don't, as in the above example. Here's why.

Sometimes ISOSURFACE generates coincident vertices - multiple vertices that are located at the same point in space. There is code in ISOSURFACE to prevent this, so this might be a bug (I'll look at it for the next release). The coincident vertices don't have too much effect on the visual appearance of the mesh, but they do confuse analysis functions like MESH_VOLUME.

But the good news is that it is easily fixed by adding:

```
result = MESH_VALIDATE(v, c, /COMBINE_VERTICES)
```

right after the call to ISOSURFACE. Now IDL reports the mesh as being solid and reports a valid volume. I don't know if this was causing your confusion or not.

For more fun, we can do an interval volume with the same volume data:

```
INTERVAL_VOLUME, vol, 5, 10, tet_verts, tet_conn
result = TETRA_CLIP([1,0,0,-20], tet_verts, tet_conn, verts_out,
conn_out)
surf_conn = TETRA_SURFACE(verts_out, conn_out)
XOBJVIEW, OBJ_NEW('IDLgrPolygon', verts_out, POLYGONS=surf_conn,
COLOR=[0,0,255])
```

This makes a sphere with a hollow center. I cut it in half with the TETRA_CLIP function and then make a surface that I can look at.

The following makes hyperbolic surfaces:

```
n = 40.0
vol = FLTARR(n,n,n)
a = (3.14159)^2 / 16.0
for i=0, n-1 do begin
  for j=0, n-1 do begin
    for k=0, n-1 do begin
      vol[i,j,k] = a * (i/n-0.5)^2 + a * (j/n-0.5)^2 +
(a-1)*(k/n-0.5)^2
```

```
    endfor
  endfor
endifor
```

```
ISOSURFACE, vol, 0, v, c
```

```
XOBJVIEW, OBJ_NEW('IDLgrPolygon', v, POLYGONS=c, COLOR=[0,0,255])
```

And finally, note that back in the sphere case, you can get a much nicer sphere if you don't round off the distances to integers. You can do this by simply changing the volume data type to float:

```
n = 40
mid = n / 2
radius = 10
vol = FLTARR(n,n,n)
for i=0, n-1 do begin
  for j=0, n-1 do begin
    for k=0, n-1 do begin
      vol[i,j,k] = SQRT( (i-mid)^2 + (j-mid)^2 + (k-mid)^2 ) + 0.5
    endfor
  endfor
endfor
```

```
ISOSURFACE, vol, radius, v, c
xobjview, obj_new('idlgrpolygon', v, polygons=c)
```

I hope that this helps you understand all this in general.

Karl

"Robert Schaefer" <robertschaefer@gmx.de> wrote in message
news:bffae64.0408180341.5223e354@posting.google.com...

- > Hello,
- > thanks for your help so far but there are some questions ;-)
- > I try to understand how Interval_volume and Isosurface works.
- > I have Problems to calculate the volume because in my real objekts
- > i do not know the radius. In Your code and my testobject we use
- > approximately a sphere. When i use other values (for example :
- > INTERVAL_VOLUME, vol, 0.1, 0.6, tet_verts, tet_conn)the results
- > are different, thats clear. But how do i know which values are
- > the right for my object to calculate the right volume?
- > I watched at the idl-example "Interval_volume" and explored the changes
- > in xobjectviev when maybe the first value set 0-> i get a box. i think
- > thats clear, because i watch at a different range. What do you think
- > can i do?
- > The same problems occure by testing with isosurface. I don't understand

> when mesh is solid and how to choose the value when there is no given
> radius or not a spherical objekt.
> (my examples: -ISOSURFACE,vol,1,v,c
> ->mesh_issolid = 1,3333
> ->mesh_volume(v,c)=6,6666
> -ISOSURFACE,vol,2,v,c
> ->mesh_issolid = 0
> -ISOSURFACE, vol, 1.9, v, c
> ->mesh_volume(v,c)=27,43
> -INTERVAL_VOLUME, vol, 0.1, 0.5, tet_verts, tet_conn
> ->tetra_volume = 23,2374)
> you described : "It depends on the range of values in the volume and
> what values you want the surface to represent." i get values by verts and
> conns (coordinates and conn-length). that might be enough to calculate
> a volume...
>
> There must be any way ;-) !?!?
>
> Robert
