Subject: Re: isosurface tetrahedra question....
Posted by Karl[1] on Fri, 04 Sep 2009 22:43:52 GMT
View Forum Message <> Reply to Message

On Sep 4, 12:21 pm, ghgm <ghgm2...@gmail.com> wrote:
> Hi there,
>
> I'm trying to create an isosurface of some 3D data and then render it
> using IDLgrPolygon.
>
> OK, so I have my 3D dataset of scalar values. At each point I have
> the X,Y, and Z coordinates which I've combined to make
> my GEOM_XYZ array....
>
> so here I go:
>
> isosurface,Data_3d,
> 0.5e-18,Outverts,Outconn,geom_xyz=GEOM_XYZ..... err, tetrahedra
> = ?
>
> I don't understand what else is needed - but apparently tetrahedra
> needs to be defined as well (another array).
>
> Can anyone please tell me what the tetrahedra array is ?
>
> Cheers
>
> George.

If you are happy with your 3d dataset (in a 3D array) implying a uniform 3D grid in space, you do not need to specify GEOM_XYZ or TETRAHEDRA.

But if your samples are from a non-uniform distribution in space, you would specify the location of each sample with a coordinate in GEOM_XYZ. Since these spatial vertices represent a possibly irregular volume and not a cube-like shape, you would describe the volume with a set of tetrahedra. Each tet would be defined by a list of 4 indices which index into your GEOM_XYZ vertex list.

The simplest possible example is:

4 samples in the data array - 1D is ok. e.g., [4,4,4,5]
4 verts that are not in the same plane into GEOM_XYZ e.g., [[0,0,0], [1,0,0],[0,1,0],[0,0,1]]
Set TETRAHEDRA to [0,1,2,3].

If you set the isovalue to 4.5, you should get a triangle back out

where all the Z coords are 0.5.

If all you are starting with is a 3d dataset of samples, then the uniform grid approach will probably suffice. It is redundant and unnecessary to generate vertices (and tet lists) if the data is on a uniform grid. Note that you can scale the returned vertices to map them from sample space to your viewing space.

Subject: Re: isosurface tetrahedra question....
Posted by ghgm2008 on Sat, 05 Sep 2009 20:54:06 GMT
View Forum Message <> Reply to Message

On Sep 4, 4:43 pm, Karl <karl.w.schu...@gmail.com> wrote:
> On Sep 4, 12:21 pm, ghgm <ghgm2...@gmail.com> wrote:
>
>
>
>> Hi there,
>
>> I'm trying to create an isosurface of some 3D data and then render it
>> using IDLgrPolygon.
>
>> OK, so I have my 3D dataset of scalar values. At each point I have
>> the X,Y, and Z coordinates which I've combined to make
>> my GEOM_XYZ array....
>
>> so here I go:
>
>> isosurface,Data_3d,
>> 0.5e-18,Outverts,Outconn,geom_xyz=GEOM_XYZ..... err, tetrahedra
>> = ?
>
>> I don't understand what else is needed - but apparently tetrahedra
>> needs to be defined as well (another array).
>
>> Can anyone please tell me what the tetrahedra array is ?
>
>> Cheers
>
>> George.
>
> If you are happy with your 3d dataset (in a 3D array) implying a
> uniform 3D grid in space, you do not need to specify GEOM_XYZ or
> TETRAHEDRA.
>
> But if your samples are from a non-uniform distribution in space, you
> would specify the location of each sample with a coordinate in

> GEOM_XYZ. Since these spatial vertices represent a possibly irregular
> volume and not a cube-like shape, you would describe the volume with a
> set of tetrahedra. Each tet would be defined by a list of 4 indices
> which index into your GEOM_XYZ vertex list.
>
> The simplest possible example is:
>
> 4 samples in the data array - 1D is ok. e.g., [4,4,4,5]
> 4 verts that are not in the same plane into GEOM_XYZ e.g., [[0,0,0],
> [1,0,0],[0,1,0],[0,0,1]]
> Set TETRAHEDRA to [0,1,2,3].
>
> If you set the isovalue to 4.5, you should get a triangle back out
> where all the Z coords are 0.5.
>
> If all you are starting with is a 3d dataset of samples, then the
> uniform grid approach will probably suffice. It is redundant and
> unnecessary to generate vertices (and tet lists) if the data is on a
> uniform grid. Note that you can scale the returned vertices to map
> them from sample space to your viewing space.

Karl,

Thanks for your help. Yes, I understand (finally) the idea that the tetrahedra defines the connections
for when the data is not on a uniform grid.

Actually, my data is on a regular grid - but it is in (r, theta , phi) coordinates.

So what I have is data_3d(256 R,30 lats,90 longs)
My X, Y and Z coords are then just (R Sin theta cos phi) etc. etc.

My question now: is r,theta,phi ok as a regular grid ? It is regular in the sense that the tetrahedra
structure is clear - but it's not uniform in the way that a standard 'cuboid' x,y,z would be.
So I'm still wondering whether ISOSURFACE will be ok with r,theta,phi as a grid or whether I have
to specify the x,y,and z for each point (and therefore the tetrahedra array as well).

I'm thinking that ISOSURFACE is probably fine with r,theta,phi - and that then your final sentence is what applies - ie,
"...you scale the returned vertices to map them from sample space to your viewing space"

so my returned vertices are in r,theta,phi space and I then just

convert to x,y,z to
pass them to IDLgrPolygon - or something ?

Subject: Re: isosurface tetrahedra question....
Posted by Karl[1] on Mon, 07 Sep 2009 23:33:58 GMT
View Forum Message <> Reply to Message

On Sep 5, 2:54 pm, ghgm <ghgm2...@gmail.com> wrote:
> On Sep 4, 4:43 pm, Karl <karl.w.schu...@gmail.com> wrote:
>
>
>
>> On Sep 4, 12:21 pm, ghgm <ghgm2...@gmail.com> wrote:
>
>>> Hi there,
>
>>> I'm trying to create an isosurface of some 3D data and then render it
>>> using IDLgrPolygon.
>
>>> OK, so I have my 3D dataset of scalar values. At each point I have
>>> the X,Y, and Z coordinates which I've combined to make
>>> my GEOM_XYZ array....
>
>>> so here I go:
>
>>> isosurface,Data_3d,
>>> 0.5e-18,Outverts,Outconn,geom_xyz=GEOM_XYZ..... err, tetrahedra
>>> = ?
>
>>> I don't understand what else is needed - but apparently tetrahedra
>>> needs to be defined as well (another array).
>
>>> Can anyone please tell me what the tetrahedra array is ?
>
>>> Cheers
>
>>> George.
>
>> If you are happy with your 3d dataset (in a 3D array) implying a
>> uniform 3D grid in space, you do not need to specify GEOM_XYZ or
>> TETRAHEDRA.
>
>> But if your samples are from a non-uniform distribution in space, you
>> would specify the location of each sample with a coordinate in
>> GEOM_XYZ. Since these spatial vertices represent a possibly irregular
>> volume and not a cube-like shape, you would describe the volume with a
>> set of tetrahedra. Each tet would be defined by a list of 4 indices

>> which index into your GEOM_XYZ vertex list.
>
>> The simplest possible example is:
>
>> 4 samples in the data array - 1D is ok. e.g., [4,4,4,5]
>> 4 verts that are not in the same plane into GEOM_XYZ e.g., [[0,0,0],
>> [1,0,0],[0,1,0],[0,0,1]]
>> Set TETRAHEDRA to [0,1,2,3].
>
>> If you set the isovalue to 4.5, you should get a triangle back out
>> where all the Z coords are 0.5.
>
>> If all you are starting with is a 3d dataset of samples, then the
>> uniform grid approach will probably suffice. It is redundant and
>> unnecessary to generate vertices (and tet lists) if the data is on a
>> uniform grid. Note that you can scale the returned vertices to map
>> them from sample space to your viewing space.
>
> Karl,
>
> Thanks for your help. Yes, I understand (finally) the idea that the
> tetrahedra defines the connections
> for when the data is not on a uniform grid.
>
> Actually, my data is on a regular grid - but it is in (r, theta , phi)
> coordinates.
>
> So what I have is data_3d(256 R,30 lats,90 longs)
> My X, Y and Z coords are then just (R Sin theta cos phi) etc. etc.
>
> My question now: is r,theta,phi ok as a regular grid ? It is regular
> in the sense that the tetrahedra
> structure is clear - but it's not uniform in the way that a standard
> 'cuboid' x,y,z would be.
> So I'm still wondering whether ISOSURFACE will be ok with r,theta,phi
> as a grid or whether I have
> to specify the x,y,and z for each point (and therefore the tetrahedra
> array as well).
>
> I'm thinking that ISOSURFACE is probably fine with r,theta,phi - and
> that then your final sentence is what applies - ie,
> "...you scale the returned vertices to map them from sample space to
> your viewing space"
>
> so my returned vertices are in r,theta,phi space and I then just
> convert to x,y,z to
> pass them to IDLgrPolygon - or something ?

I *think* it would work as you want.

ISOSURFACE performs simple linear interpolation along each of the three dimensions to determine where the isosurface passes between two samples. As long as your 3d system is linear, it should work. ISOSURFACE really isn't aware of the units or meaning of the three dimensions.

When given just the 3d data, the sample coordinates are implicit. In this case, the "grid" will run from 0-255, 0-29, and 0-89. If the isovalue is 0.8 and the sample at [34,0,0] is 0.9 and the sample at [35,0,0] is 0.7, then the coordinate of one of your returned vertices will be [34.5,0,0].

Now I doubt that your actual r's, lat's and long's are actually 0-255, 0-29, and 0-89, but you can probably get from these implicit coords to your actuals via a simple linear transform. Further, the "shape" of these samples in space is not going to be a cube, since the points with smaller r's are going to be closer together. Still, I think that ISOSURFACE's interpolation will be OK.

So, yes, I think that you would need to apply linear transforms to the implicit grid coords returned by ISOSURFACE (scale + offset) to get to the area where the samples really are, and then convert from r sin theta to XYZ.

Subject: Re: isosurface tetrahedra question....
Posted by ghgm2008 on Tue, 08 Sep 2009 17:46:22 GMT
View Forum Message <> Reply to Message

On Sep 7, 5:33 pm, Karl <karl.w.schu...@gmail.com> wrote:
> On Sep 5, 2:54 pm, ghgm <ghgm2...@gmail.com> wrote:
>
>
>
>> On Sep 4, 4:43 pm, Karl <karl.w.schu...@gmail.com> wrote:
>
>>> On Sep 4, 12:21 pm, ghgm <ghgm2...@gmail.com> wrote:
>
>>>> Hi there,
>
>>>> I'm trying to create an isosurface of some 3D data and then render it
>>>> using IDLgrPolygon.
>
>>>> OK, so I have my 3D dataset of scalar values. At each point I have
>>>> the X,Y, and Z coordinates which I've combined to make
>>>> my GEOM_XYZ array....

>
>>>> so here I go:
>
>>>> isosurface,Data_3d,
>>>> 0.5e-18,Outverts,Outconn,geom_xyz=GEOM_XYZ..... err, tetrahedra
>>>> = ?
>
>>>> I don't understand what else is needed - but apparently tetrahedra
>>>> needs to be defined as well (another array).
>
>>>> Can anyone please tell me what the tetrahedra array is ?
>
>>>> Cheers
>
>>>> George.
>
>>> If you are happy with your 3d dataset (in a 3D array) implying a
>>> uniform 3D grid in space, you do not need to specify GEOM_XYZ or
>>> TETRAHEDRA.
>
>>> But if your samples are from a non-uniform distribution in space, you
>>> would specify the location of each sample with a coordinate in
>>> GEOM_XYZ. Since these spatial vertices represent a possibly irregular
>>> volume and not a cube-like shape, you would describe the volume with a
>>> set of tetrahedra. Each tet would be defined by a list of 4 indices
>>> which index into your GEOM_XYZ vertex list.
>
>>> The simplest possible example is:
>
>>> 4 samples in the data array - 1D is ok. e.g., [4,4,4,5]
>>> 4 verts that are not in the same plane into GEOM_XYZ e.g., [[0,0,0],
>>> [1,0,0],[0,1,0],[0,0,1]]
>>> Set TETRAHEDRA to [0,1,2,3].
>
>>> If you set the isovalue to 4.5, you should get a triangle back out
>>> where all the Z coords are 0.5.
>
>>> If all you are starting with is a 3d dataset of samples, then the
>>> uniform grid approach will probably suffice. It is redundant and
>>> unnecessary to generate vertices (and tet lists) if the data is on a
>>> uniform grid. Note that you can scale the returned vertices to map
>>> them from sample space to your viewing space.
>
>> Karl,
>
>> Thanks for your help. Yes, I understand (finally) the idea that the
>> tetrahedra defines the connections
>> for when the data is not on a uniform grid.

>
>> Actually, my data is on a regular grid - but it is in (r, theta , phi)
>> coordinates.
>
>> So what I have is data_3d(256 R,30 lats,90 longs)
>> My X, Y and Z coords are then just (R Sin theta cos phi) etc. etc.
>
>> My question now: is r,theta,phi ok as a regular grid ? It is regular
>> in the sense that the tetrahedra
>> structure is clear - but it's not uniform in the way that a standard
>> 'cuboid' x,y,z would be.
>> So I'm still wondering whether ISOSURFACE will be ok with r,theta,phi
>> as a grid or whether I have
>> to specify the x,y,and z for each point (and therefore the tetrahedra
>> array as well).
>
>> I'm thinking that ISOSURFACE is probably fine with r,theta,phi - and
>> that then your final sentence is what applies - ie,
>> "...you scale the returned vertices to map them from sample space to
>> your viewing space"
>
>> so my returned vertices are in r,theta,phi space and I then just
>> convert to x,y,z to
>> pass them to IDLgrPolygon - or something ?
>
> I *think* it would work as you want.
>
> ISOSURFACE performs simple linear interpolation along each of the
> three dimensions to determine where the isosurface passes between two
> samples. As long as your 3d system is linear, it should work.
> ISOSURFACE really isn't aware of the units or meaning of the three
> dimensions.
>
> When given just the 3d data, the sample coordinates are implicit. In
> this case, the "grid" will run from 0-255, 0-29, and 0-89. If the
> isovalue is 0.8 and the sample at [34,0,0] is 0.9 and the sample at
> [35,0,0] is 0.7, then the coordinate of one of your returned vertices
> will be [34.5,0,0].
>
> Now I doubt that your actual r's, lat's and long's are actually 0-255,
> 0-29, and 0-89, but you can probably get from these implicit coords to
> your actuals via a simple linear transform. Further, the "shape" of
> these samples in space is not going to be a cube, since the points
> with smaller r's are going to be closer together. Still, I think that
> ISOSURFACE's interpolation will be OK.
>
> So, yes, I think that you would need to apply linear transforms to the
> implicit grid coords returned by ISOSURFACE (scale + offset) to get to

> the area where the samples really are, and then convert from r sin
> theta to XYZ.

Karl,

Thanks again for your help..... I've now had time to try it all out
- and yes, your previous post is correct - at least the isosurface
that
it creates looks fine.

So, yes, the isosurface routine finds vertices in terms of the indexes
of the r,theta,phi data.
The actual r,theta phi positions for each vertex can then be
calculated
(by linear interpolation where a fraction, like your '34.5' shows up).

From these we can transform to x,y,z and then send this to the
IDLgrpolygon routine.

The connectivity (Outconn) remains untouched.

So this is what I do:

isosurface,D_data_3d,0.5e-18,Outverts,Outconn

Outverts2 = Outverts converted from indexes (ir,ilat,ilon) to actual
r,theta,phi

Outverts3 = Outverts2 converted from r,theta,phi to x,y,z

green_object = OBJ_NEW('IDLgrPolygon', Outverts3 , polygons =
Outconn , color=[0,230,0])

cheers,

George.