Notes on calculating a VOI with E4D

Leigh, ExploreGeo 12 March 2021

This document goes through explaining how I've come to get a VOI/DOI (Volume of Investigation/ Depth of Investigation) when using E4D.

So far, only the real component works.

Real Conductivity/Resitivity

Starting from the values for average conductivity, or a choice from what looks more likely to be the final resistivity from a fully run model, run a model from halfspaces at 10x higher and 10x lower conductivities.
You can stop each model after 5 iterations. From these, compute both:

$$\mathrm{Log}(\rho_H) - \mathrm{Log}(\rho_L)$$

and

$$(\rho_H - \rho_L)/(\rho_{HS}-\rho_{LS})$$

where ρ_H and ρ_L are the values of resistivity (not conductivity) at the given iteration from the high starting model, and ρ_{HS} and ρ_{LS} are the starting halfspace values for higher and lower respectively.

To do this easily (until Kim creates a better method), copy and rename the sigma files from E4D appropriately (ie sigma.1 from the low directory => CND_LOW_1.sig, etc). Grab a copy of the e4d_VOI_calc directory from Pet100_Raid/Code/E4D. Copy the renamed sig files to this directory. To use the program voi_calc within this directory, compile with ./go.sh, and then call the program with the following 4 arguments for each iteration:
> ./voi_calc [filename_low] [filename_high] [0 for res, 1 for phase calculations] [0 for pos/neg, 1 for abs, optional, defaults to 0]
So for us this would be:
> ./voi_calc [filename_low] [filename_high] 0

Copy the file generated back to an appropriate directory, such as the one with the mesh it was generated from, and run bx to join the create a .exo viewable in paraview.
This exodus file has two attributes when created from res mode voi_calc:
real_conductivity
This attribute is the ExploreGeo VOI:

$$\mathrm{Log}(\rho_H) - \mathrm{Log}(\rho_L)$$

complex_conductivity
This attribute is an Oldenburg and Li (1999) method from "Estimating depth of investigation in dc resistivity and IP surveys", Geophysics:

$$(\rho_H - \rho_L)/(\rho_{HS}-\rho_{LS})$$

Unfortunately I haven't gotten around to allowing custom names in exodus, but it should be an easy change with another optional argument
Now perform the following for each iteration:

- Open file in paraview

- Load in real_conductivity, complex_conductivity, and block 1 only.
- Right click on the data in paraview>filter>Cell data to point data
- Create a clip that starts clipping right at one of the sides of the viewable mesh, such that at the moment it doesnt clip anything
- Now load in block 2
- Color with a linear EG Rainbow stretch for both real_conductivity and complex_conductivity
- Create an Isosurface of the model with a linear stretch for your desired metric (EG VOI or Oldenburg and Li VOI). This isosurface will let you visualise the distribution easily
- Create another isosurface at about 0.08. If this isosurface is clearly wrong, try a different number, or a different iteration.

From my limited experience, the best outcome comes from iteration 4 with an EG isosurface of about 0.08. If this is clearly wrong, try a different iteration and/or a different isosurface. Sometimes setting the color stretch to be equally form positive to negative with a colour scale white in the middle may help isolate problems.

You can then output this single isosurface to a obj and then externally convert it to a dxf for viewing in other programs a client may have.

Phase/Chargeability

Repeat the above but from a regularly finished real mode, with the starting models **phase** being 10x higher and lower. To do this simply replace each complex_conductivity value in the sigma file of a finished iteration with the average phase*10 or /10, then multiplied by the real_conductivity value. So if a cell has a value of $1*10^{-3}$ and the starting models average phase was 5mRad=0.005 Rad, the complex_conductivity value for the HIGH model will be:

$$(1*10^{-3})*(0.005*10)=5*10^{-5}$$

and the LOW model will be

$$(1*10^{-3})*(0.005/10)=5*10^{-7}$$

These make use of the small angle approximation, but if you take the arc tan of the result it should be about equal to your average phase*10 or /10.

You could do this with a quick Fortran routine, a quick python routine, or simply copying the data into excel, making a formula for the phase and copying it back (it should be able to handle even a dataset of ~1,000,000 points)

Additionally, make sure to use the phase option in voi_calc:

```
> ./voi_calc [filename_low]  [filename_high] 1
```

to properly use the output real and complex conductivities from E4D.

The values functions output are the same as for Res/Real conductivity, simply with resistivity replaced by phase.

Again, iteration 4 appears to be the best, with a value of about 0.08 being correct, but your mileage may vary.