

On the visualization of simulations

Francesca Poli, Princeton Plasma Physics Laboratory

The following is the wish list of a user, with very poor knowledge of software engineering and data integration.

If I have to imagine the ideal visualization interface, I would picture it as a graphic interface with a combination of scrolling menus, typing windows and plotting windows.

If I were to look at the results of a TRANSP simulation, for example, or any other time-dependent transport code, or even data from an real experiment (why not?) I would then be able to select the runID and the plasma parameter I need to plot.

The interface should allow me to look at both the time-dependent evolution as well as the 2D surface profile and individual profiles at selected times. I imagine I would select the time of interest from a rolling bar at the bottom of the window that reports the time-dependent data, for example total plasma current.

I should be able to look both at completed simulations and interim results.

The interface should include also the magnetic equilibrium data. If the simulation is predictive, it might be helpful to be able to compare input and output for differences.

Data can be accessed either from a NETCDF file or from a tree-structure, like it is for the MDS trees or for the under-development IMAS at the ITER Organization.

The type of graphic interface described above can probably be used for visualization of simulation output from more complex codes, like MHD or turbulence simulations. In this case, I imagine that there is a need for outputs from codes to be in a standard format as well, more importantly to be of 'manageable' size. I let experts in the field to comment what quantities make sense to plot and visualize and under what form, either contour plots of potential or growth rates of individual modes.

I imagine I would like to save data in a file, which might be for example, an .eps figure for presentation or an ASCII file with the (x,y) data that I can then plot separately, combined with other quantities.

The ElVis visualization program, developed by Eliot Feibush at PPPL, is an example of initiative taken in this direction.

It has many of the desirable features and capabilities described above. Unfortunately, it has not been adequately promoted among users. The limited distribution and usage has compromised opportunities for development and software improvement that is usually driven by user request. One example is processing speed, another is graphic quality.

The ElVis project is worth pursuing forward, by first advertising among potential users, maybe a pool of volunteers, and moving forward based on the feedback of these testers.

Perhaps interaction between this Panel and Panel G (Software Integration and Performance) lead to innovative ideas for improved algorithms for visualization?