

# 1 Flux Wrapper

This section integrates the CELL and FLUX codes in a very simple way. The idea was to take the program from the preceding week, and add a single button to its main configuration menu for setting up the flux calculation. The previous project involved changing the cell code so that it would create LabWindows buttons and controls. Unlike the cell calculation the original FLUX.EXE file was used, and the FLUX.C program was not modified. The cell program was used as a base for setting up the flux configuration file and launching the flux program.

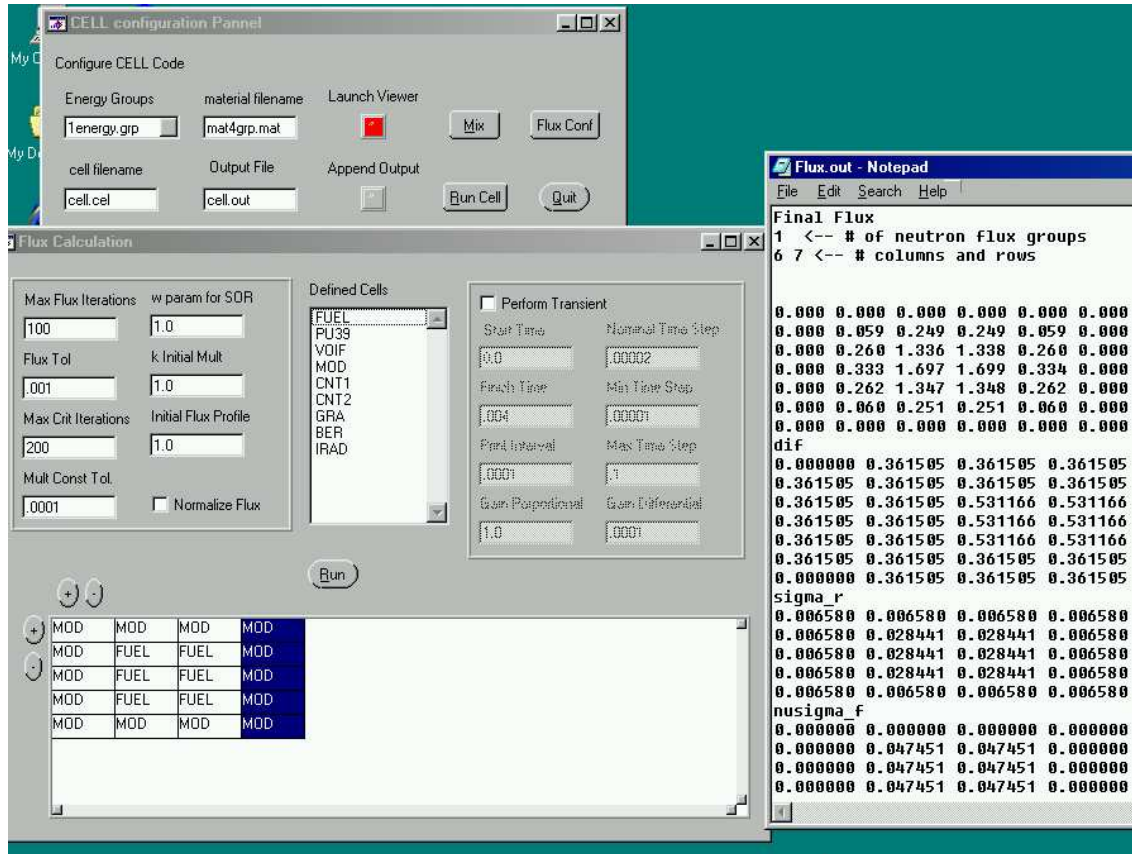


Figure 1: Example Flux Session

The flux configuration window is divided into four main areas. The first area in the upper left lists the control parameters for the steady state calculation. In the upper right, the control parameters for the transient calculation are entered. Since the flux code depends on the cell code, the when the flux configuration window is opened it first runs the cell code using the existing parameters from the main cell menu. The middle field lists the mixtures available to the flux code which result from running the cell code.

The bottom area of the flux configuration window allows cells to be arranged. The user adds and subtracts rows from the cell region by clicking on the plus and minus buttons in the upper left of the region. Cells are added as blanks and must be defined by the user. Individual cells are defined by first selecting an area in the cell region with the mouse, and then clicking on one of

the cell types from the defined list. The flux.out file is displayed in a pop-up window after the program has finished.

## 2 Comments

This version is filled with bugs, too many to name. The effort was made to illustrate the range of functionality, rather than supply a polished program. Launching FLUX.EXE as a DOS program creates several difficulties. Firstly the program waits for a key-press before it terminates which doesn't fit well with the LabWindows environment. Secondly running the program in the background complicates data sharing and multi-threading issues discussed in the previous paper. The positive issue illustrated here is that in the case where source code is not available, and only the executable is present it is still possible to integrate the executable into the overall project, albeit in a rather ad-hoc way.

This project as it is demonstrates the basic functionality of combining the CELL and FLUX programs treating the CELL program as the base, and interface for both. The overall organization isn't very good at this point, and the suggestion is made that for the next session, the existing software is cleaned up, debugged, and documented, without adding significant functionality.

### 2.1 Comments Regarding LabWindows

Last week the question was raised as to whether LabWindows was a suitable development environment. I discussed the question with 3 people, all who had some experience with Visual programming under windows. According to them MS Visual C++ is far too complicated, and based on their descriptions I tended to agree. From the same conversations my impression is that Visual Basic is probably too simple, and since it isn't compiled would lack the necessary efficiency for some of the simulation programs.

Although a tool like Delphi (Visual Pascal) might provide a more structured and straightforward programming environment, I think that the advantages of the link with the DAQ drivers that LabWindows supplies outweighs this. C is generally a more sophisticated language than Pascal, and this has both advantages and disadvantages. I have found that the LabWindows compiler seems quite reliable and good at detecting errors, to date Windows hasn't crashed on me while running the LabWindows compiler.

My suggestion is to stick with the LabWindows compiler, but my cautionary comment is that I think some students will need a large chunk of time to familiarize themselves with it. At this point I estimate I have logged 50-70 hours with this compiler, and I'm still learning things about it, and often confused by its responses. I find the internal documentation to be sufficient, but I have 8 years worth of C programming experience- I believe a person with less experience will at least need more tutorial style materials.