

## UIUC's CONTRIBUTION TO BIG-10's INIE PROJECT

Yuxiang Gu, Nick Karancevic, Koji Sugawara, Yizhou Yan, James Stubbins, and  
Rizwan-uddin\*

*Department of Nuclear, Plasma and Radiological Engineering  
University of Illinois, 103 S. Goodwin Ave., Urbana, IL 61801 USA*

Recent progress toward assembling and testing a set of codes for the design of a research reactor is reported. Graphical user interfaces are developed for ORIGEN and DIF3D. KENOIV is used to optimize the beam design of a TRIGA reactor (reported in a companion paper). Results of recent development of a virtual environment for a radiation lab as well as for a URTR control room are also reported.

**KEYWORDS:** *INIE Project, URTR, Virtual Reality, GUIs, Codes*

### 1. Introduction

University of Illinois at Urbana-Champaign is taking the lead in identifying key attributes of an advance university research and training reactor (URTR) as a first step toward the design and construction of a new URTR. This is the second task in the list of four major tasks identified for the Big-10 consortium partnering under the INIE program. A major component of this task is to take advantage of virtual reality [1-5] in the design and development stages of this URTR. This will lead to a virtual URTR that will play an integral part in the other three tasks of the broader project. Specifically, it will be a critical component in the efforts toward: 1) the development of innovative programs in graduate research and undergraduate education and training; 2) the creation of a novel educational outreach and education program for the nuclear industry and national laboratories; and 3) facilitating collaborative research among the consortium universities, DOE laboratories and other institutions. University of Illinois is collaborating with other partner universities and institutions in accomplishing these tasks.

Efforts are currently focused along four initially parallel directions that have already started and will completely merge toward the latter stages of this project. These are:

1. Design and assemble a computational environment for in-core, near-core and beam line analysis.
2. Develop I/O interfaces and visualization environment for the computational tools.
3. Develop a virtual model of a URTR facility (pool, core, control room, beams, ...)
4. Identify desirable attributes (specs) in a future URTR.

As part of item 3, in addition to the task identified above, progress has also been made toward the development of a virtual model of a "radiation laboratory." When completed this will provide a novel educational outreach tool as well as a stepping-stone for the much more complicated task of developing a virtual URTR.

This paper is divided into two parts. Recent developments in I/O interfaces and progress toward assembling a set of visualization environments are reported in the next section. Progress toward a virtual radiation lab as well as a virtual URTR are reported in Section 3. Recent contributions toward task number 1 are discussed in a companion paper.

---

\* Corresponding author, Tel. 217-244-4944, E-mail: rizwan@uiuc.edu

## 2. Codes and Interfaces

Among others, the set of codes selected for design of a URTR are DIF3D [6,7], ORIGEN [8,9], Monteburn, SCALE4.4 system, and MCNP. We have recently developed graphical user interfaces (GUI) for ORIGEN and DIF3D.

### 2.1 GUI for ORIGEN

ORIGEN is a well-known burnup code. There are several parameters and flags in ORIGEN2 input file, *tape5.inp*. Large quantity of data is written in the output files of ORIGEN2. A GUI for ORIGEN2 has been developed using Visual Basic. *Explorer* style windows are developed to help user easily manage all input and output files of ORIGEN2. A picture of the ORIGEN2 GUI is shown in Fig. 1. Left box shows various input/output files that can be opened in the right box. An example of an input file, *tape5.inp*, is shown in the right box. In application for input file development pop-up help functions are also added. *Compile* and *Run* options are also available on the interface.

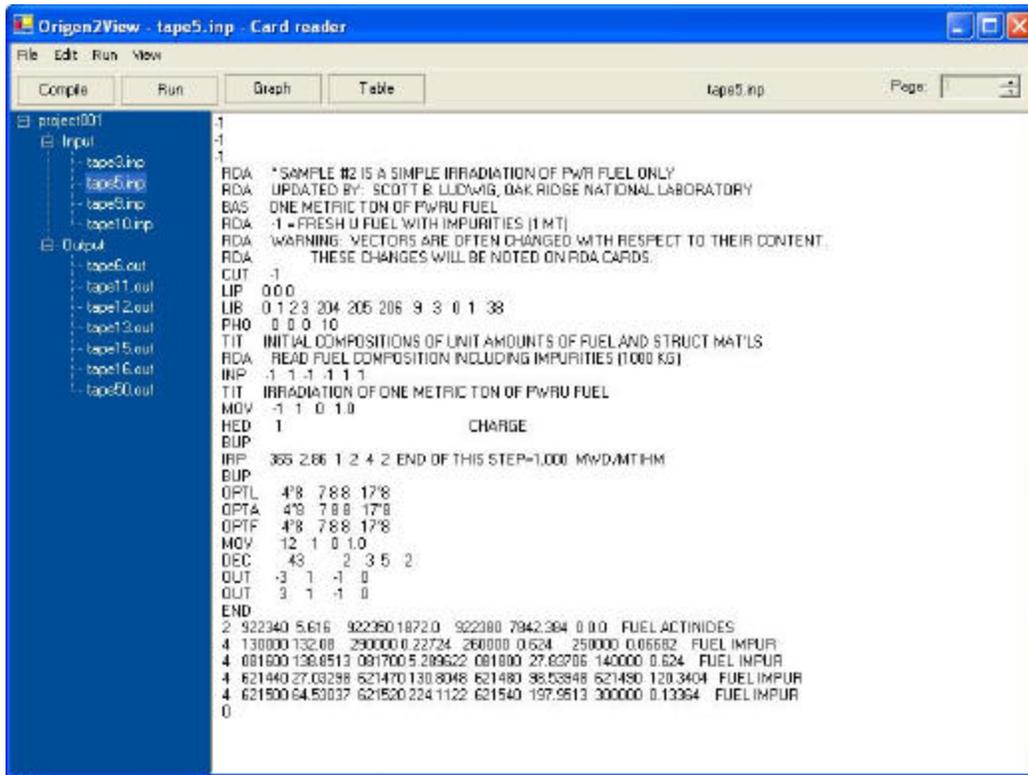
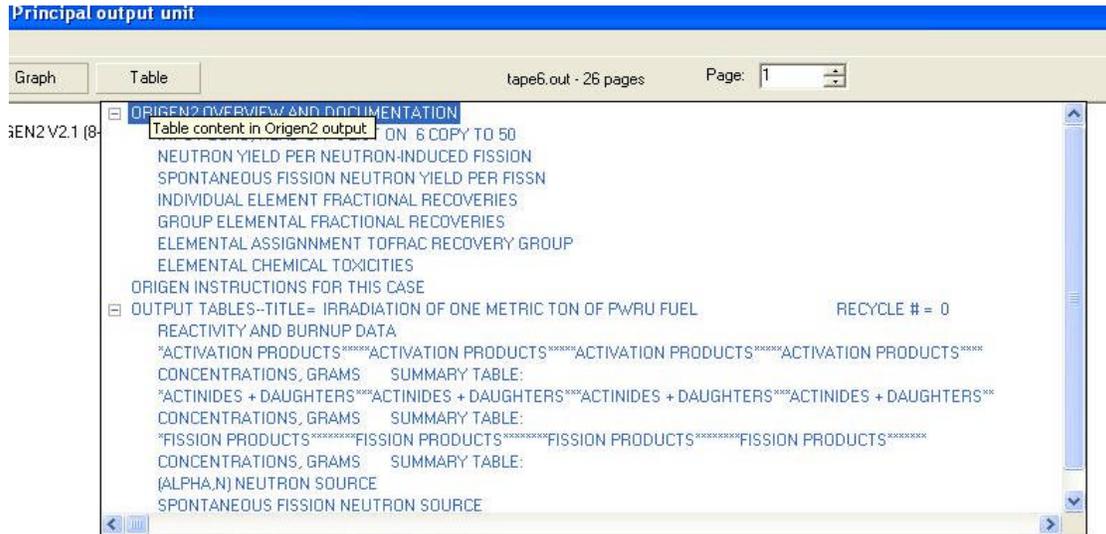


Fig.1 GUI for ORIGEN2 showing *tape5.inp*.

Functions for post processing of results generated by ORIGEN2 are made available in a separate window. Application extracts data tables from output file and shows the index of tables in a list, so that user can reach these tables by just clicking on the appropriate entry. The user also can plot the data in these tables. It can be done fairly easily by simply clicking on the data table to be plotted. The user can then print or save these graphs as bitmap file. An example of the window for post processing of

ORIGEN2 data is shown in Fig. 2.



**Fig.2** GUI for ORIGEN2 showing an index of output tables.

## 2.2 GUI for DIF3D

DIF3D is a well-known multi-group neutron diffusion code. GUI for DIF3D was also developed using Visual Basic as a *Windows* application. Since DIF3D runs on workstations running UNIX or PCs running LINUX operating system, the application is divided into two parts: a front end and a back end.

### *DIF3D Front End*

A text file is used as an interface between the front end and DIF3D's input format. The original input format for DIF3D, especially the cross-sections data, is in fixed format that makes its use a little cumbersome. In the front end developed here, an interactive graphical interface allows a user to design a reactor visually. User can draw geometry, set boundary conditions, and choose various flags (options) easily by clicking the mouse. Geometry (x, y, z), dimensions, number of nodes and node size can be specified by the user. A checkerboard style color model of the core (and reflector) is displayed by the application. Different slices in the z-direction can be viewed. Figure 3 shows an example of this screen. A box on the right allows material selection for each node (click-click). Box for boundary conditions and mesh size selection is shown in Fig. 4.

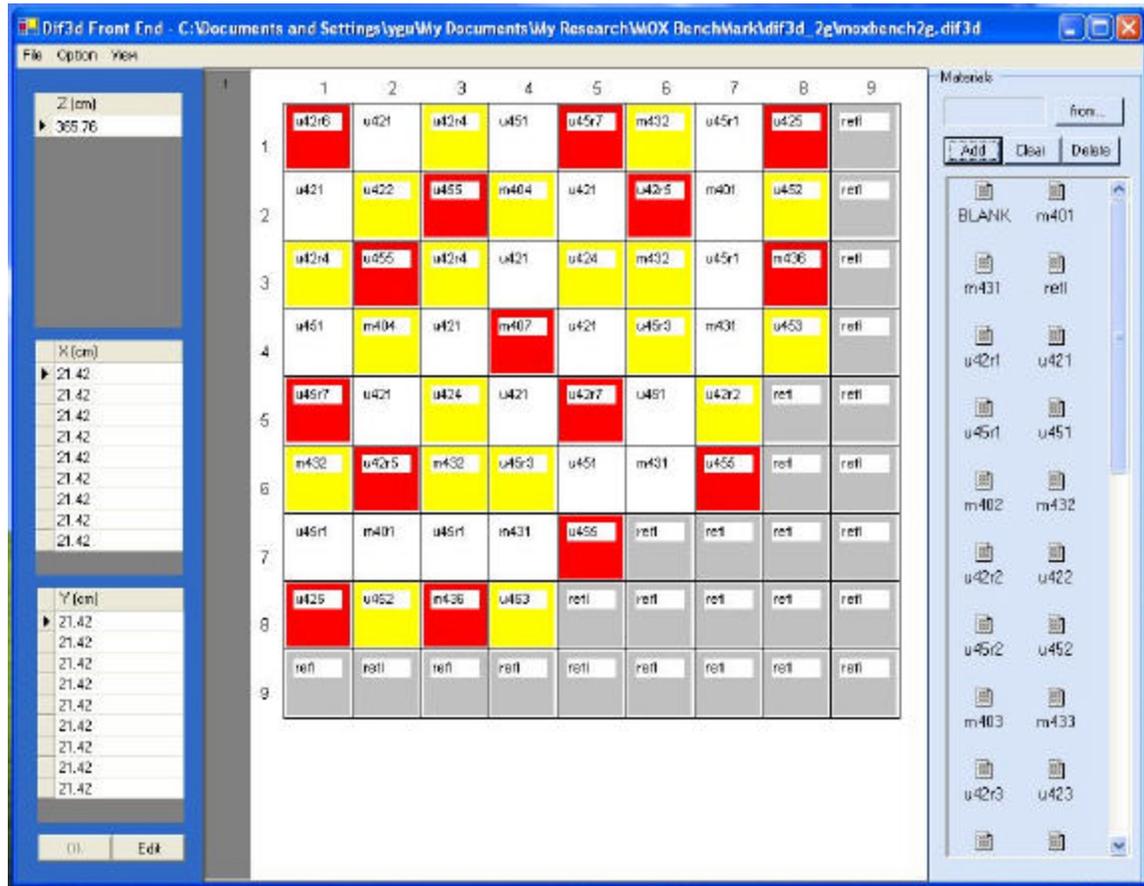


Fig.3 DIF3D front end; interactive graphical interface for geometry and material selection.

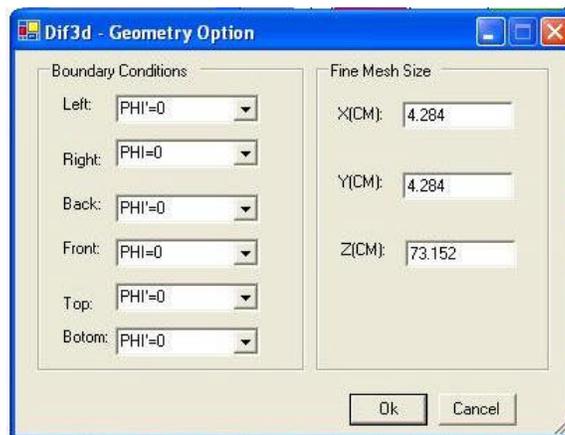
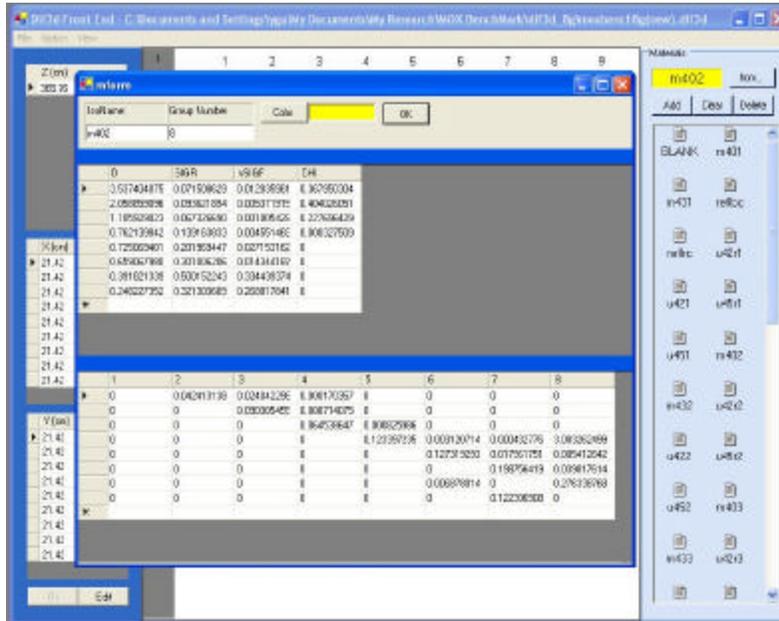
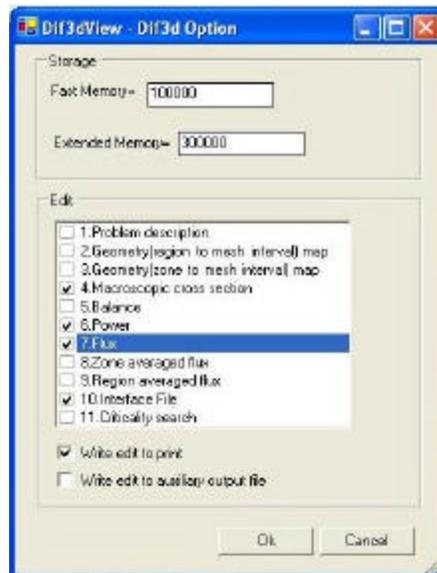


Fig.4 DIF3D front end; geometry and boundary conditions options.

Data grid control was used so that user can input cross section data into table, without having to worry about format. It can also import cross section from spreadsheets, like Excel, in a specified format. An example of a box for group specific material properties is shown in Fig. 5. It allows the user to easily change any material property. An example box for selection of various parameters and flags is shown in Fig. 6.



**Fig.5** DIF3D front end; for cross section specification for each material.



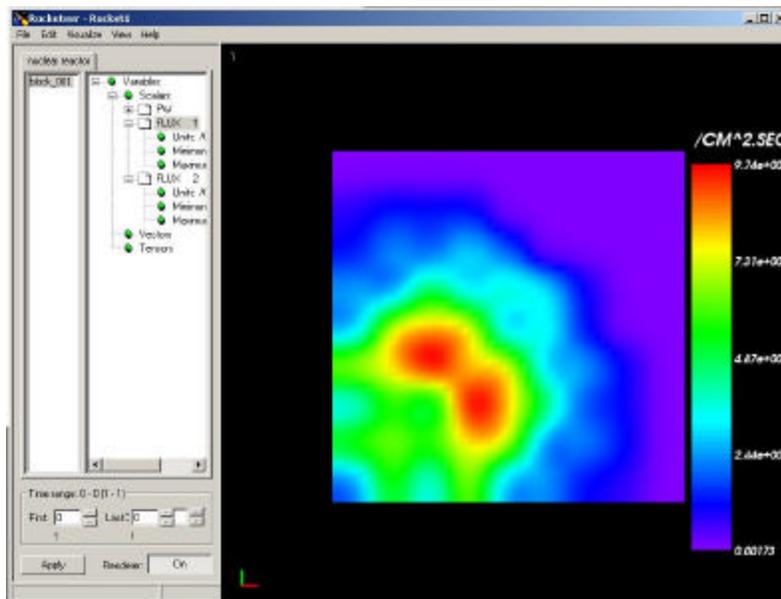
**Fig.6** DIF3D front end; for various options.

Once the geometry and nodes are built, boundary conditions specified, material for each node selected, and all flags and parameters have been specified, the application can transform the information into the BCD input format for DIF3D. This file is then ready to be used as the input file for a DIF3D run.

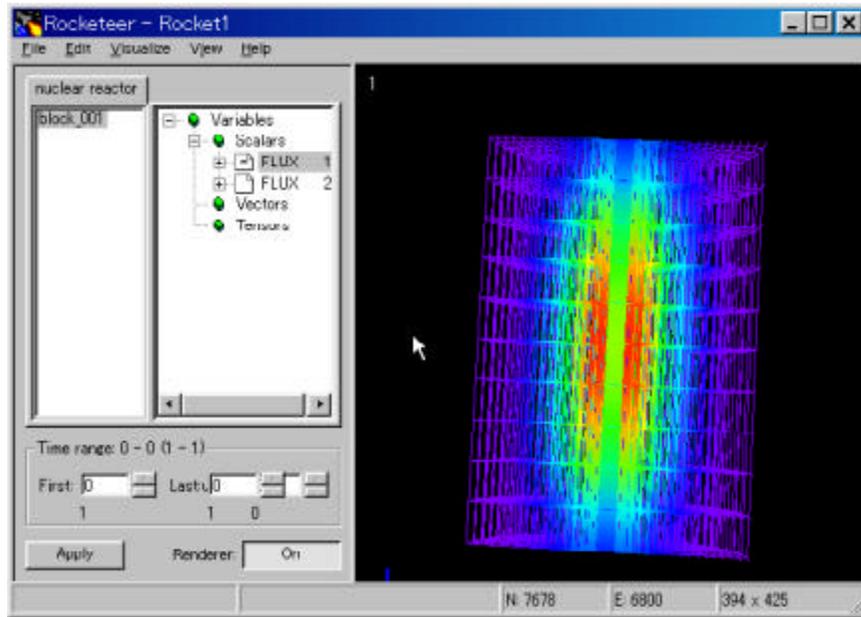
### DIF3D Back End

Due to the large amount of 3D data generated by DIF3D, a start-from-scratch approach for post-processing is not feasible. Instead, post-processing applications available in public domain were explored to identify one suitable for our purpose. We identified and selected a code called ROCKETEER [10] for post processing purposes.

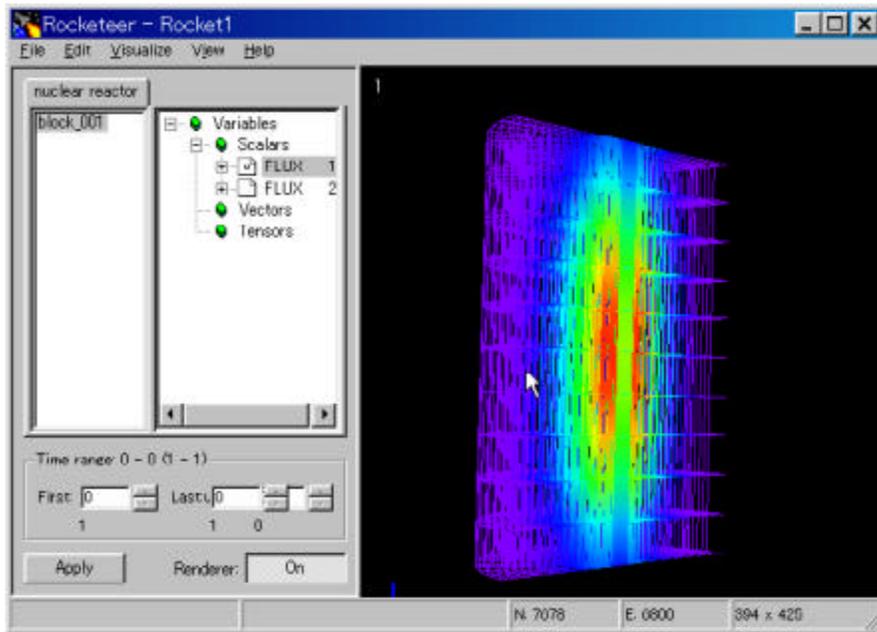
*Rocketeer* is a 3D visualization tool for numerical data sets. *Rocketeer* can show a set of data as mesh, solid surface, and glyphs using a color scale. It can also display iso-surfaces and/or slices across x, y, or z axes. *Rocketeer* input should be in HDF (stands for Hierarchical Data Format) format. HDF [11] is a set of software and file formats for scientific data management developed by NCSA [12]. The HDF software includes I/O libraries and tools for analyzing, visualizing, and converting scientific data. *Rocketeer* can be used to visualize flux and/or power distribution data generated by DIF3D. For the purpose, the DIF3D output data need to be converted into HDF4 format. Hence, for the back end, a code was written to transform the binary output of DIF3D, including the data and geometry information into HDF4 format using Rocketeer API. The HDF4 files, that act as interface between DIF3D output and *Rocketeer*, can then be readily used with *Rocketeer*. Figure 7 shows a *Rocketeer* window showing the fast flux in a plane of a reactor core. Figures 8 and 9 show the flux in a 3D wiremesh geometry of the cylinder. Using a cutting plane, only half of the cylinder is visualized. The half cylinder can be rotated easily using the mouse (Figs. 8 and 9).



**Fig. 7** Backend post processing of DIF3D generated data using *Rocketeer*. Neutron flux in a plane.



**Fig.8** Backend post processing of DIF3D generated data using Rocketeer. Neutron flux in a wiremesh.

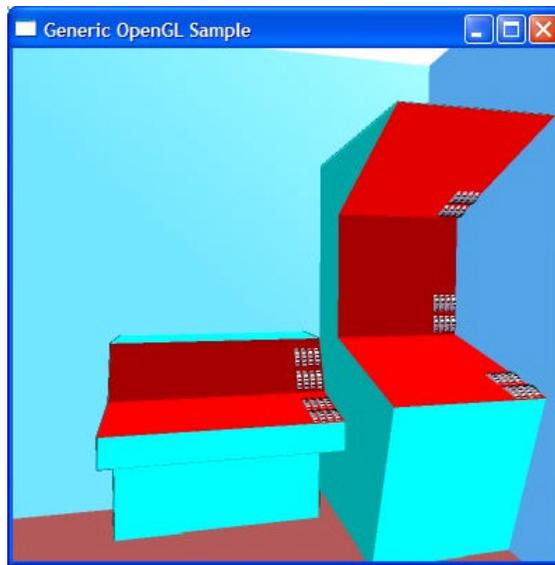


**Fig.9** Backend post processing of DIF3D generated data. Neutron flux in a wiremesh (rotated).

### 3. Virtual Reality

Virtual reality is an excellent tool for education, outreach, and for research purposes [1]. Its use by nuclear scientists and engineers is also on the rise [2-4]. Reported herein is recent progress made towards creation of virtual nuclear facilities varying from simple experiments to a complete reactor site. This part of the project is focused on adding enhanced capabilities to the basic computer code, ranging from standard graphics routines, to hierarchical and organizational data structures. In this stage of the project, the available models are confined to simple radiation experiments, models, and concepts, such as those seen in a typical undergraduate curriculum.

Numerous video games, with ever increasing levels of realism, are developed continuously using simple tools such as the C and C++ programming languages, and various low-level 3D graphics libraries. Due to cross-platform compatibility, OpenGL is the graphics library of choice. Using the aforementioned programming languages and graphics library allows for code execution on the common Windows PCs, as well as on the many flavors of Linux, Unix, and Irix (used to power the NCSA [12] virtual reality facility known as the CAVE). This setup also allows for complete flexibility, great power, and the ability to easily incorporate any needs specific to nuclear simulations. An example of a simple control room development in early stages is shown in Fig. 10.



**Fig. 10.** A simple control room in early stages of development.

#### 3.1 Virtual Models

Exclusive of trivial models, there are currently three interactive simulations. In the first, the user can hold a virtual Geiger counter in her hand, and walk around in a (virtual) radiation field due to a point radiation source located behind a shield. The Geiger counter displays the radiation level (from a pre-calculated database) at its current location (Fig. 11a). The second virtual experiment consists of measuring the half-life of a radioactive substance. In this case, the virtual detector simply displays the temporal resolution of the number of counts. The data displayed by the detector can be either pre-computed in a database, or calculated real-time (Fig. 11b). The third interactive simulation is a model of the GUS (graphite uranium sub-critical [assembly]) facility at the University of Illinois. Driven by a source, this sub-critical assembly is used for undergraduate laboratories involving

measurements of flux distribution, buckling, neutron leakage, and so on. A virtual model of this facility – still under development – provides a tour of the laboratory, and will have the capability to conduct virtual experiments in the future (Fig. 11c). Figure 11d shows a realistic picture of the virtual radiation lab with some control panels, multi-channel analyzers, and the GUS facility.

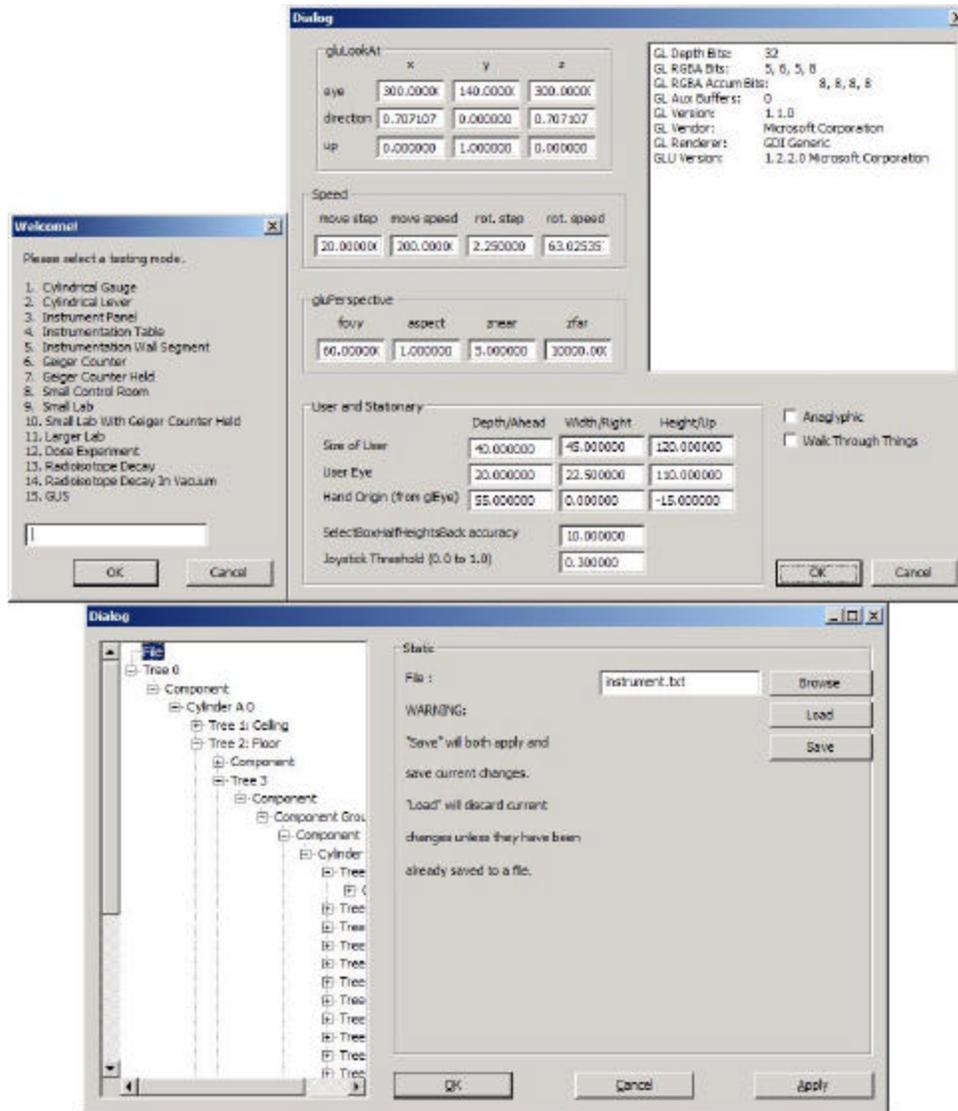
The importance of hiding intricate details of programming is crucial in the development of these virtual



**Fig. 11** (Clockwise from top-left) a) Shielding experiment; b) Decay experiment; c) GUS measurements; and d) A test site, with control panels, a desk and some chairs, GUS, and some multi-channel analyzers.

models as with any engineering process. An interactive capability, utilizing standard Windows dialog boxes, has thus been developed to shield the last layer of model developers from the “messy” details. Three dialog boxes of this “developer’s tool” are shown in Fig. 12. The dialog box in Fig. 12a lists pre-defined models available for execution, while the one shown in Fig. 12b contains various options and debugging parameters, including the use of a 3D anaglyphic display. The dialog box shown in Fig. 12c

is the most powerful one, allowing for on-the-fly changes to any component of the simulation, as well as for saving and loading any model files.



**Fig. 12** (Clockwise from top-left) a) The choice of models available; b) General and debug options; and c) VR model editor.

Codes have also been written that allow a user to display numerical data from 3D simulations, such as those from DIF3D, in the CAVE. A generic capability has been developed that allows data written in a class of HDF-4 standard files to be easily displayed in the CAVE. At this time only limited manipulations (rotation, translation, cutting planes, etc) have been incorporated. In addition, another conversion routine has been written that allows *vrml* files to be displayed in the CAVE. Models created in CAD software can be saved in the *vrml* format, and then easily displayed in a virtual environment after passing through the above described code.

#### 4. Summary

Substantial progress has been made toward the goals of the INIE project. Windows applications have been developed to facilitate the use of popular codes such as ORIGEN2 and DIF3D. They provide a user friendly interface to these codes. For post processing of DIF3D's 3D data, a data visualization code called Rocketeer has been selected. Since Rocketeer works with a standard HDF4 format data files, a code has been written to convert the binary output of DIF3D into HDF4 format. This data can also be displayed in the CAVE.

In addition to these, virtual models have been developed for nuclear radiation lab as well as for a nuclear reactor control room. Simple virtual experiments can be conducted, and the model can also be used for outreach purposes.

#### References

- 1 ) W.R. Sherman, A.B. Craig, Understanding Virtual Reality, Elsevier Science, San Francisco, CA (2003).
- 2 ) N. Karancevic, Rizwan-uddin, "Virtual Systems for Understanding and Advancement of Nuclear Power," ANS/ENS International Winter Meeting, New Orleans, LA (2003).
- 3 ) V. E. Whisker, M.E. Warren, A.J. Baratta, T. S. Shaw, "Development of a Radiation Dose Model for Immersive Virtual Environments," ANS/ENS International Winter Meeting, New Orleans, LA (2003).
- 4 ) Rizwan-uddin, N. Karancevic, S. Tikves, "Virtual Reality: At the Service of GEN-IV, and V, and .....", "Proceedings of ICAPP, Cordoba, Spain (2003).
- 5 ) National Center of Supercomputing Application (NCSA); Visualization group website: <http://cave.ncsa.uiuc.edu/>
- 6 ) K. L. Derstine, "DIF3D: A Code to Solve One-, Two-, and Three-Dimensional Finite-Difference Diffusion Theory Problems," ANL-82-64 (1984).
- 7 ) R. D. Lawrence, "The DIF3D Nodal Neutronics Option for Two- and Three-Dimensional Diffusion Theory Calculations in Hexagonal Geometry," ANL-83-1 (1983).
- 8 ) A. G. Croff, "ORIGEN2 Code Package CCC-371," Informal Notes (October 1981).
- 9 ) A. G. Croff, "A User's Manual for the ORIGEN2 Computer Code," ORNL/TM-7175 (July 1980).
- 10) Center for Simulation of Advanced Rocket, University of Illinois; Rocketeer web site: [http://www.csar.uiuc.edu/F\\_software/rocketeer/](http://www.csar.uiuc.edu/F_software/rocketeer/)
- 11) National Center of Supercomputing Application (NCSA); HDF group <http://hdf.ncsa.uiuc.edu/>
- 12) National Center for Supercomputing Applications (NCSA) web site, <http://www.ncsa.uiuc.edu>