



# **International Agreement Report**

---

## **Nowadays Tools for Graphical Post-Processing of TRAC-BF1 Results**

Prepared by

P. Corcuera, M. Garcés

Applied Mathematics and Computer Science Department  
University of Cantabria  
SPAIN

Office of Nuclear Regulatory Research  
U.S. Nuclear Regulatory Commission  
Washington, DC 20555-0001

October 2000

Prepared as part of  
The Agreement on Research Participation and Technical Exchange  
under the International Code Application and Maintenance Program (CAMP)

Published by  
U.S. Nuclear Regulatory Commission

## AVAILABILITY OF REFERENCE MATERIALS IN NRC PUBLICATIONS

### NRC Reference Material

As of November 1999, you may electronically access NUREG-series publications and other NRC records at NRC's Public Electronic Reading Room at [www.nrc.gov/NRC/ADAMS/index.html](http://www.nrc.gov/NRC/ADAMS/index.html).

Publicly released records include, to name a few, NUREG-series publications; *Federal Register* notices; applicant, licensee, and vendor documents and correspondence; NRC correspondence and internal memoranda; bulletins and information notices; inspection and investigative reports; licensee event reports; and Commission papers and their attachments.

NRC publications in the NUREG series, NRC regulations, and *Title 10, Energy*, in the Code of *Federal Regulations* may also be purchased from one of these two sources.

1. The Superintendent of Documents  
U.S. Government Printing Office  
P. O. Box 37082  
Washington, DC 20402-9328  
[www.access.gpo.gov/su\\_docs](http://www.access.gpo.gov/su_docs)  
202-512-1800
2. The National Technical Information Service  
Springfield, VA 22161-0002  
[www.ntis.gov](http://www.ntis.gov)  
1-800-533-6847 or, locally, 703-805-6000

A single copy of each NRC draft report for comment is available free, to the extent of supply, upon written request as follows:

Address: Office of the Chief Information Officer,  
Reproduction and Distribution  
Services Section  
U.S. Nuclear Regulatory Commission  
Washington, DC 20555-0001

E-mail: [DISTRIBUTION@nrc.gov](mailto:DISTRIBUTION@nrc.gov)

Facsimile: 301-415-2289

Some publications in the NUREG series that are posted at NRC's Web site address [www.nrc.gov/NRC/NUREGS/indexnum.html](http://www.nrc.gov/NRC/NUREGS/indexnum.html) are updated periodically and may differ from the last printed version. Although references to material found on a Web site bear the date the material was accessed, the material available on the date cited may subsequently be removed from the site.

### Non-NRC Reference Material

Documents available from public and special technical libraries include all open literature items, such as books, journal articles, and transactions, *Federal Register* notices, Federal and State legislation, and congressional reports. Such documents as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings may be purchased from their sponsoring organization.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at—

The NRC Technical Library  
Two White Flint North  
11545 Rockville Pike  
Rockville, MD 20852-2738

These standards are available in the library for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from—

American National Standards Institute  
11 West 42<sup>nd</sup> Street  
New York, NY 10036-8002  
[www.ansi.org](http://www.ansi.org)  
212-642-4900

The NUREG series comprises (1) technical and administrative reports and books prepared by the staff (NUREG-XXXX) or agency contractors (NUREG/CR-XXXX), (2) proceedings of conferences (NUREG/CP-XXXX), (3) reports resulting from international agreements (NUREG/IA-XXXX), (4) brochures (NUREG/BR-XXXX), and (5) compilations of legal decisions and orders of the Commission and Atomic and Safety Licensing Boards and of Directors' decisions under Section 2.206 of NRC's regulations (NUREG-0750).

**DISCLAIMER:** This report was prepared under an international cooperative agreement for the exchange of technical information. Neither the U.S. Government nor any agency thereof, nor any employee, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this publication, or represents that its use by such third party would not infringe privately owned rights.

NUREG/IA-0190



# International Agreement Report

---

## Nowadays Tools for Graphical Post-Processing of TRAC-BF1 Results

Prepared by

P. Corcuera, M. Garcés

Applied Mathematics and Computer Science Department  
University of Cantabria  
SPAIN

Office of Nuclear Regulatory Research  
U.S. Nuclear Regulatory Commission  
Washington, DC 20555-0001

October 2000

Prepared as part of  
The Agreement on Research Participation and Technical Exchange  
under the International Code Application and Maintenance Program (CAMP)

Published by  
U.S. Nuclear Regulatory Commission

**NUREG/IA-0190 has been  
reproduced from the best available copy.**

## Abstract

For the time being, there are no standard tools for graphical post-processing of TRAC-BF1 results. A new specialized application based on PC/Windows technology, GraTRAC, has been developed to integrate in a unique tool the best capabilities found in other post-processing packages. It simplifies the access to the output variable of interest by means of a very intuitive graphical interface and launches Xmgr5 to offer the whole functionality of this tool in treating data and plots. Besides, the user is allowed to customize the graphical interface with a simple procedure in order to implement a representation of the nodalization of the simulated model. A first release is being used by our group to analyze data from the last TRAC simulations. The users feelings are very encouraging.

On the other hand, a clear tendency to improve T/H codes with the inclusion of detailed three-dimensional models and by linking them to 3-D neutronic codes is observed. We have developed VisVas3D, a prototype application based on the simple TRAC-BF1 Vessel component, able to display 3-D information in a dynamic and interactive way. Running on the PC/Windows environment it serves to demonstrate the enormous possibilities that specialized tools can offer in this field.

The same programming approach has been used in the development of both projects: the latest advances in object oriented technologies combined with the utilization of well-known and easy to access third party tools to deal with the harder task of graphics generation and rendering. This solution has proved very efficient resulting in a fast development.



## Table of Contents

<b>1. Introduction</b> .....	<b>1</b>
<b>2. GrafTRAC, an Integrated Graphical Post-processor</b> .....	<b>4</b>
2.1 Design Considerations .....	5
2.2 Application Architecture .....	6
2.3 GrafTRAC Software Components: NodoPlus and TextoPlus .....	8
2.4 Program Functionality .....	10
2.5 Creating Custom Made Interface .....	12
2.6 Future Work .....	15
<b>3 VisVas3D, a Dynamic and Interactive Visualization Prototype</b> .....	<b>16</b>
3.1 Design Considerations .....	17
3.2 Application Architecture .....	18
3.3 Program Functionality .....	19
3.4 Future Work .....	21
<b>4 Conclusions</b> .....	<b>23</b>
<b>REFERENCES</b> .....	<b>24</b>





## **Executive Summary**

This report presents two new software tools designed to provide a better insight into the output data produced by TRAC-BF1. They also help the user to analyze and handle that information in a friendly and efficient manner using standard current technologies on the PC/Windows environment. The first one is an integrated graphical post-processor which covers all the analysis tasks, from the easy access to output variables to the generation of elaborated plots ready for printing by means of the recognized Xmgr5 program. The other is a prototype application in the area of three-dimensional data visualization. It makes use of a quite simple 3-D model included in TRAC-BF1 to explore the possibilities that this kind of tools can offer when applied to T/H codes. It allows a dynamic and interactive visualization of the output data. We may affirm that the design objectives of both applications have been achieved to a very high degree.

This work constitutes the first contribution of the University of Cantabria to the Code Application and Maintenance Program leaded by US NRC.



# 1 Introduction

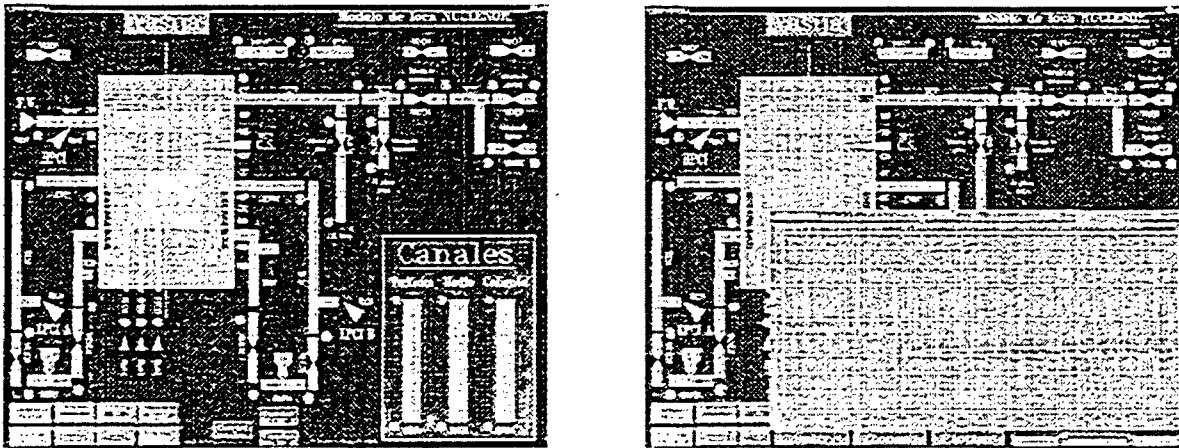
Our group joined the Spanish CAMP community just two years ago. However, we have been working with T/H codes for many years, mainly in the framework of continued agreements between the University of Cantabria and Nuclenor S.A., the owner of Sta. María de Garoña NPP [1].

From this long experience we have learned, as every user of this sort of codes, that visualization plays a major role in the process of analyzing the enormous quantities of generated output data. In spite of that, there are no standard tools, at present, to carry out this job. Each group has been compelled to manage its own solution, more or less effective, to the visualization problem. Furthermore, the necessity of running these resource-consuming codes in powerful workstations distinct from the usual desktop environment does not contribute to facilitate this task. The recent OECD/CSNI workshop on advanced thermal-hydraulic and neutronic codes held in Barcelona this year recognized that, even now, more effort is required in the visualization area.

In the special case of TRAC-BF1 the situation gets worse since the NRC attempts to develop software for an efficient use of the codes have been centered in other tools: RELAP first and the future consolidated code TRAC-M now. This fact has been corroborated in the development of the Nuclear Plant Analyzer [2] and the new Symbolic Nuclear Analysis Package [3][4]. There is no doubt that the latter, SNAP, will solve many of the current problems but, by now, there is just one first and limited release to work with RELAP5 and there are no plans to extend its capabilities to other codes. Thereby, until TRAC-M development has been concluded and made available to all users, and that could take some years, TRAC users will be forced to conform with their old visualization tools.

The first software applications presented here, the integrated graphical post-processor GrafTRAC, is intended to serve as the present-day response to the visualization requirements of TRAC-BF1 users. After having used many packages to develop custom utilities capable of

displaying TRAC-BF1 results, some of them shareware as the X-Window Forms library [5] or the GnuPlot program [6] and some proprietary as DataViews [7], we have taken advantage of all this experience for GrafTRAC development. Figure 1 shows the aspect of our most advanced previous tool that has been used until now on Unix platforms. As detailed later, GrafTRAC integrates the best features of the rest of visualization and analysis packages able to deal with TRAC-BF1. It profits from the outstanding capabilities of Xmgr5, the NRC extension of the general shareware visualization package Xmgr [8][9].



*Figure 1. Two images of the main window of a previous graphical post-processor for TRAC-BF1 developed by our group. On the left side showing the model nodalization and on the right with a generated plot that compares TRAC data to real plant data.*

The other application introduced in this work, VisVas3D, is more ambitious but at the moment offers less functionality. Its development has been inspired from the irreversible tendency to incorporate more and more complex three-dimensional models into the T/H codes or link them to neutronic 3-D codes. At present, VisVas3D is just a prototype application based on a quite simple 3-D model. It has been designed to show the advantages that specialized tools can offer when working with this type of models and also the difficulties found in its development; not to carry out a precise analysis of code output data.

Today, there are many and very varied tools conceived to visualize and analyze three-dimensional information, although almost any of them are specially dedicated to T/H or neutronic codes. From the graphics libraries that can be used directly in the development of applications, to the integral visual environments that allow to create and visualize complex 3-D graphics by means of flux diagrams. Besides, there are packages designed to represent 3-D information in VRML format (Virtual Reality Modeling Language), the standard language for the transmission of 3-D images over the Internet. References [10] and [11] make an exhaustive study of many of these tools with detailed descriptions.

Assuming the increased competition and decreased cost of PC/Windows technology, that allow the use of standard, off-the-shelf software and hardware for functions that, in the past, were performed only by proprietary, specialized hardware and software, we have built VisVas3D as a Windows based application. As will be explained in chapter 3, the large amount of information involved in 3-D images generation and rendering can be easily managed by standard applications as VisVas3D running on today desktop computers.

## 2 GrafTRAC, an Integrated Graphical Post-processor

GrafTRAC is not “yet another” plotting software for thermal-hydraulic codes. It has been designed to integrate, in one single application, the most useful capabilities found while working with different tools devoted to conventional graphics generation, i.e., bi-dimensional graphics of the temporal evolution of one or more variables. Given our previous experience and our present necessities in the field of T/H simulation, the tool has been specifically conceived to serve as graphical post-processor for the TRAC-BF1 code. However, its functionality could be easily extended to visualize and analyze results from different codes.

The main features of GrafTRAC, detailed below, are the simple access to the desired code output variable through a very intuitive graphical interface, the powerful capacities offered by Xmgr5 to manage data and plots and the possibility of modifying the primary interface in a simple manner. In some way, GrafTRAC could be described as an specialized, improved and customizable Xmgr5 graphical interface for the TRAC-BF1 code. At the moment, it runs on PC platforms as a Windows native application. The performance of today common desktop computers with Windows operating systems has demonstrated its capability to deal with the huge TRAC output files in a satisfactory way. However, X-Window emulation software is also required because the plots are visualized through Xmgr5.

The development effort was highly reduced by using advanced object oriented programming techniques. The software reusability concept was broadly exploited. Two specialized software components, Active-X controls as described later, were developed in order to simplify the application architecture and help to easily create the program main windows which present the nodalization of the analyzed case. Our group has gained great experience on this programming approach that has shown its suitability in much more complex projects [12].

Furthermore, the use of Xmgr5 for displaying plots and handling data has also simplified the development of GrafTRAC. Being Xmgr5 users, we know well the excellent features of this

tool, very difficult to improve or even equal with an existing or new application with regard to these kind of tasks. Besides, it has been distributed among TRAC users, so it is not necessary to include it in the setup program of our application.

Recently, we have released the first version of GrafTRAC. Since then, our group is using it to analyze the results from the last TRAC simulations of Sta. Maria de Garoña NPP. Although this release is little more than a beta version, the users' feeling is very encouraging. Many points of improvement have been identified and will be undertaken for the following versions. By now, the Spanish version is available to the TRAC-BF1 users community in our country.

## **2.1 Design Considerations**

The main goal for GrafTRAC development was to serve as the only graphical post-processor for the TRAC-BF1 code. Thus, it must respond adequately to all user demands with respect to data analysis and plot generation. Moreover, it had to make these tasks efficient and easy to undertake.

In order to reach these goals the following design considerations were taking into account:

- The application must execute on PC platforms since they are the common equipment accessible to all users. Thereby, the obtained plots could be easily inserted into documents in electronic format.
- The graphical interface of the program must simplify, as much as possible, the access to the desired output variable among thousands found in a typical TRAC-BF1 execution. It is not enough with an ordered list as that offered by Xmgr5.
- It must be assured that any of the output variables can be accessed through that interface.
- A simple method to create and modify the primary interface should be provided so the user may represent the nodalization of any TRAC input deck.
- The user must be able to open several files at the same time and select variables from different files that can be plotted in the same graph.

- It would be interesting to include modules for loading data from other simulation codes in a way the tool could be used to establish comparisons between them and TRAC-BF1. Besides, a module to read general ASCII column data files could also be incorporated and used to load real plant data.
- The maximum functionality regarding raw data transformation and analysis was required. It was considered that Xmgr5 capabilities were the best available option. Only some simple tasks in this area could be carried out directly from the primary GrafTRAC interface, mainly those related to common units change.
- The same could be said regarding plot generation and configuration. Xmgr5 suffices to accomplish this job. In addition, Xmgr5 provides the required printing functionality to obtain hardcopies of the plots or insert graphs in other documents.

Apart from these specific features, the application had to be developed keeping in mind the rules of good GUIs (graphical user interfaces) programming: friendliness, robustness, standard ways of interaction and so on.

## 2.2 Application Architecture

Looking for the best performance when executing GrafTRAC on Windows platforms, we have used the Microsoft's Visual C++ programming environment [13] during the development of the whole application. The object oriented programming techniques are perfectly suited for the construction of this type of packages and easier to apply to Windows programs by means of the recognized Microsoft's library of classes MFC (Microsoft's Foundation Classes) and the more recent component software technology. In addition, Visual C++ guarantees a greater speed of execution over other more popular programming tools like Visual Basic.

The primary interface of GrafTRAC is based on two custom-made and highly specialized ActiveX components. The next section explains briefly this technology and how it has been applied to the development of the controls.



Apart from this feature, GrafTRAC is a classic Windows application with document/view architecture. It includes all the features expected in a Windows interface including menu bar, accelerator keys, status bar, standard open file dialogs and so on. The main window is designed to show the nodalization of the TRAC model being analyzed. The interface is completed with an additional nodalization window, which could serve to display some particular section of the model, fuel channels for instance, and a series of dialog boxes with lists of the open files and the available and selected variables.

Respect to the modules that provide GrafTRAC functionality, the main are those responsible for loading data and creating the input file for Xmgr5 which contains the complete description of the plot that is going to be visualized. There is a specific module for searching the available variables in each data file according to user's selections. Once a variable is found, the position of the corresponding values in the data file is saved for subsequent fast access but data is not loaded yet. This operation only takes place after the user decides to visualize a plot and that variable is among the selected ones. The necessary data files are read again and the required values actually loaded. Then, a general module writes commands and data in the adequate format into a temporary ASCII file so that Xmgr5 understands it to create the desired graph. When this file is closed, Xmgr5 is launched and, if an X emulation package is running, the plot appears on the PC screen.

In addition to these principal modules, GrafTRAC contains a large set of simpler functions devoted to handle files and variables, show or hide dialog boxes which exchange data, provide some extra utilities that will be mentioned in section 2.4 and, in general, respond to user interaction in an satisfactory way.

In order to simplify GrafTRAC distribution, an standard setup program for Windows platforms (Windows 95, 98 and NT) has been developed. It installs the GrafTRAC application (a default version with a model of Sta. Maria de Garona NPP), the GrafTRAC controls NodoPlus and TextoPlus, a Visual C++ pre-built project to allow users to create their own interfaces as explained later and a TRCGRF file to be used as an example of the application features.

## 2.3 GrafTRAC Software Components: NodoPlus and TextoPlus

Two visual ActiveX components provide the basic functionality of the primary interface of GrafTRAC. This Microsoft's component technology uses the standard Component Object Model (COM) to enable software components, or controls, to interact with one another in one networked environment, regardless the language in which they were created. Usually they are developed with Visual C++ to be inserted in Visual Basic forms. As explained above, we preferred to use Visual C++ for the whole process. Regarding future improvements, there are two interesting features of ActiveX technology. One, it can be used to create applications that run on the Internet besides the desktop. Two, there are several available tools to convert them to Java controls, the famous 'java beans', which can be integrated directly in Java applications.

The ActiveX controls (formerly known as OLE controls) are just small in-process servers that can be used in any OLE container application. Thus, each control is a reusable software object that supports a wide variety of OLE functionality and can be easily customized to fit our needs. Its appearance and behavior is defined by means of stock and custom properties and methods, similar in use and purpose, respectively, to member variables and member functions of a C++ class. The control container, i.e. GrafTRAC in our case, communicates with the object accessing to these properties and methods. The ActiveX control also sends messages, 'fire events' in the ActiveX jargon, to notify the container that something important has happened in the control, usually a change of state or an user action.

With regard to GrafTRAC functionality the essential control is NodoPlus. It has been designed to represent one node of any TRAC-BF1 component in the application interface. Its properties, specified when a customized interface is being developed as described in section 2.5, include type of TRAC component and node identification (component index, corresponding level or rod group, if needed, and cell number). In addition, the user can introduce a brief text description and select the text fonts used to display the component index. The main property page of NodoPlus is shown in figure 2.

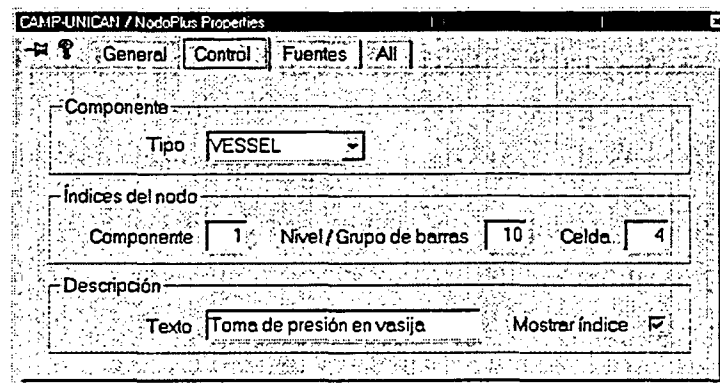


Figure 2. Main property page of the GrafTRAC ActiveX component *NodoPlus*.

The control paints itself with a predefined background color assigned according to the type of component. In response to user action it shows an informative dialog box with its identification and description when clicked. If double clicked, it sends a message to the container application that includes the node identification and remains highlighted for some seconds. When GrafTRAC receives this message, it can show the list of associated output variables found in the active data file.

The other GrafTRAC ActiveX control, *TextoPlus*, is a merely descriptive component. It permits to add text in any place of the GrafTRAC interface in order to include useful permanent information or just to decorate the nodalization image. The user can select the desired fonts, text color and background color.

## 2.4 Program Functionality

When the application is launched, GrafTRAC shows the main nodalization window along with a dialog box, the 'selection dialog', containing two list boxes where the open data files and the selected variables will be added. This dialog includes buttons to close data files, erase variables from the list and visualize the plot with the evolution of the selected variables. These can correspond to different data files (five can be open simultaneously) with a maximum of ten variables to be included in the same plot. Figure 3 displays the main windows of GrafTRAC with a model of Sta. Maria de Garoña NPP.

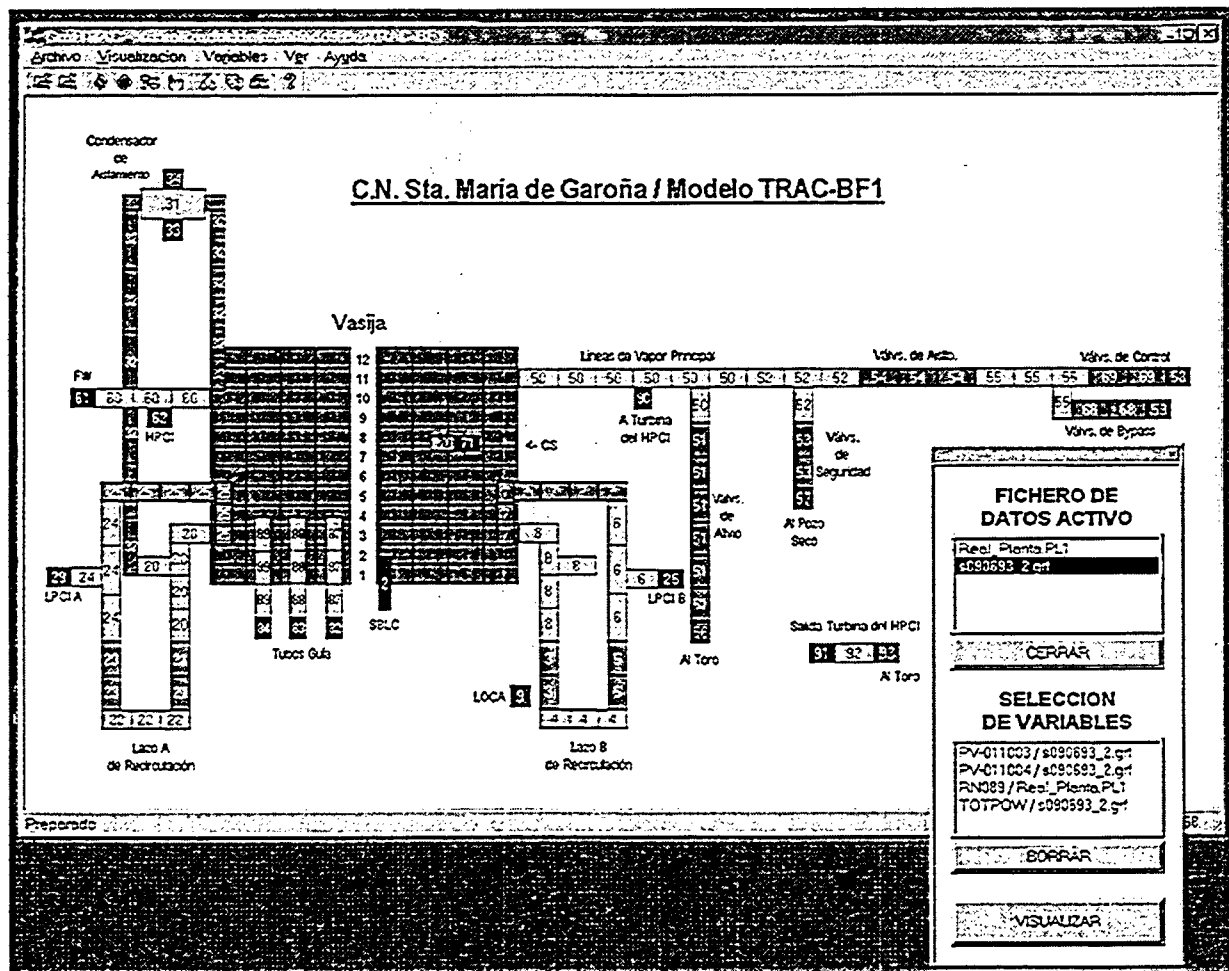


Figure 3. GrafTRAC main windows. The TRAC model corresponds to Garoña NPP.

Regarding the accepted data files, this first version of GrafTRAC is capable of reading (TRAC file option) standard binary TRCGRF files and also the ASCII 'ftn08' files that are obtained as graphic output in some TRAC installations. The ASCII data file option allows the user to load an ASCII file with data in columns (first line with variable names and first column with time data). This option has been extended to read output files from MAAP (file extension '.PL\*'), the popular severe accident code frequently used by our simulation group. When a file has been open, it is added to the 'selection dialog' list. Clicking on it makes that file the active one.

When the active file is a TRAC output file the access to variables can be done by double clicking the NodoPlus components included in the nodalization windows. Figure 4 shows an example of the extra nodalization window representing a core model. As explained above, a dialog box will display a list with the associated variables found in the file and, when selected, the variable will be added to the corresponding 'selection dialog' list. Those variables that are not directly linked to any node can be reached through the global variables option (time and calculation related variables, core power, water level in vessel...) or through the control variables option (output from the control blocks). The 'choose variable' option permits a fast search of the desired variable introducing its TRAC name. When the active file is not a TRAC file, its variables can only be accessed through the global variables option (a list with all the variables found in the file is displayed) or directly through the 'choose variable' option.

The rest of menu choices serve to launch the extra nodalization window, to show the 'selection dialog' in case it had been closed, to establish some units conventions that will apply to all the visualized plots and, finally, to load an Xmgr5 parameter file that will be used in the generation of plots. This feature is very useful since it allows, for instance, to create an specific mask for all the plots of the same sequence where the scenario is described with Xmgr5 titles or texts.

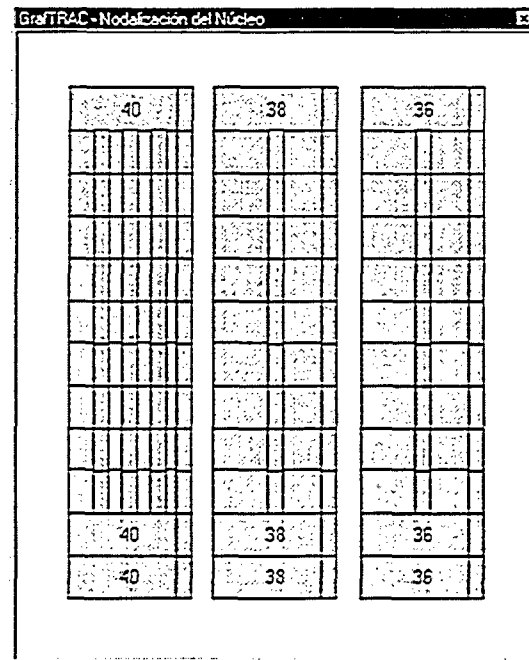


Figure 4. GrafTRAC additional nodalization window with Garoña NPP core model.

When the 'Visualize' button has been pressed and Xmgr5 launched showing the desired graph, the user may employ the whole Xmgr5 functionality to analyze data, transform it, customize the plot or obtain hardcopies. This job, usually the last in the post-processing course, can be accomplished with excellent results in any common PC making use of the Xmgr5 PostScript features and of any of the shareware packages available to deal with this kind of files (we use Aladdin's Ghostscript with Ghostgum's GSView). In figure 5 we show a Xmgr5 plot as appears after having been launched by GrafTRAC with a parameter file loaded.

## 2.5 Creating Custom Made Interfaces

One of GrafTRAC main features is that the user is able to build or modify a customized interface easily. The package is distributed with a pre-built Visual C++ project that he can use to modify the nodalization windows by means of the GrafTRAC controls. Thereby a new GrafTRAC application that represents his own TRAC model will be obtained. The Resource Editor utility included in the Visual C++ environment makes this task quite simple. No

programming skills of any kind are needed. The only requirement is to have access to a Microsoft Visual C++ 6.0, or later, license. We assume that this is not a true restriction since the package is shipped with Microsoft Visual Studio, currently the most common programming suite for Windows platforms.

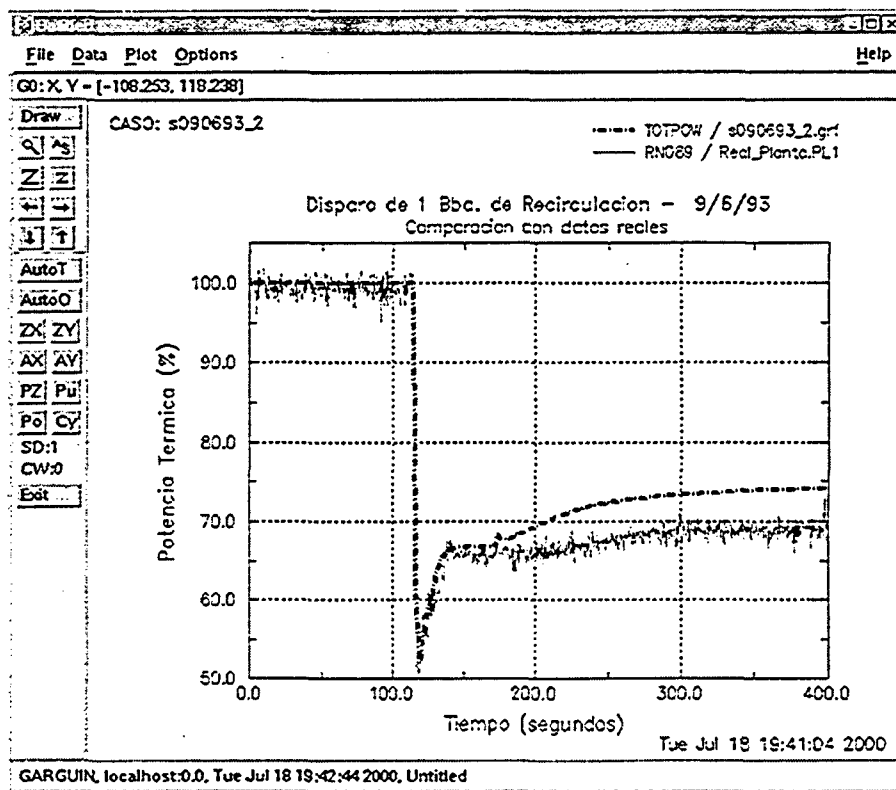


Figure 5. Xmgr5 plot obtained from GrafTRAC with a parameter file loaded.

Once a copy of the pre-built GrafTRAC project is open (it is convenient to conserve the original copy of the project), the user is only allowed to edit two of the dialogs included in the resource file. They correspond to the main and additional GrafTRAC nodalization windows. Using the drag and drop method he can insert NodoPlus and TextoPlus controls from the Resource Editor's controls bar onto these dialogs. Then, he can modify the properties of these controls by means of the provided property pages. Moreover, the controls are completely sizable and can be aligned, copied or cut as desired to obtain a satisfying interface. Figure 6 shows a typical image of the process.

When both windows have been completed, the user has to specify a distinctive name for the executable file and rebuild the project. The new GrafTRAC version produced will maintain all the application functionality without any additional programming effort. Following this procedure, as many as needed different GrafTRAC versions can be created. The same method applies to the modification of existing versions.

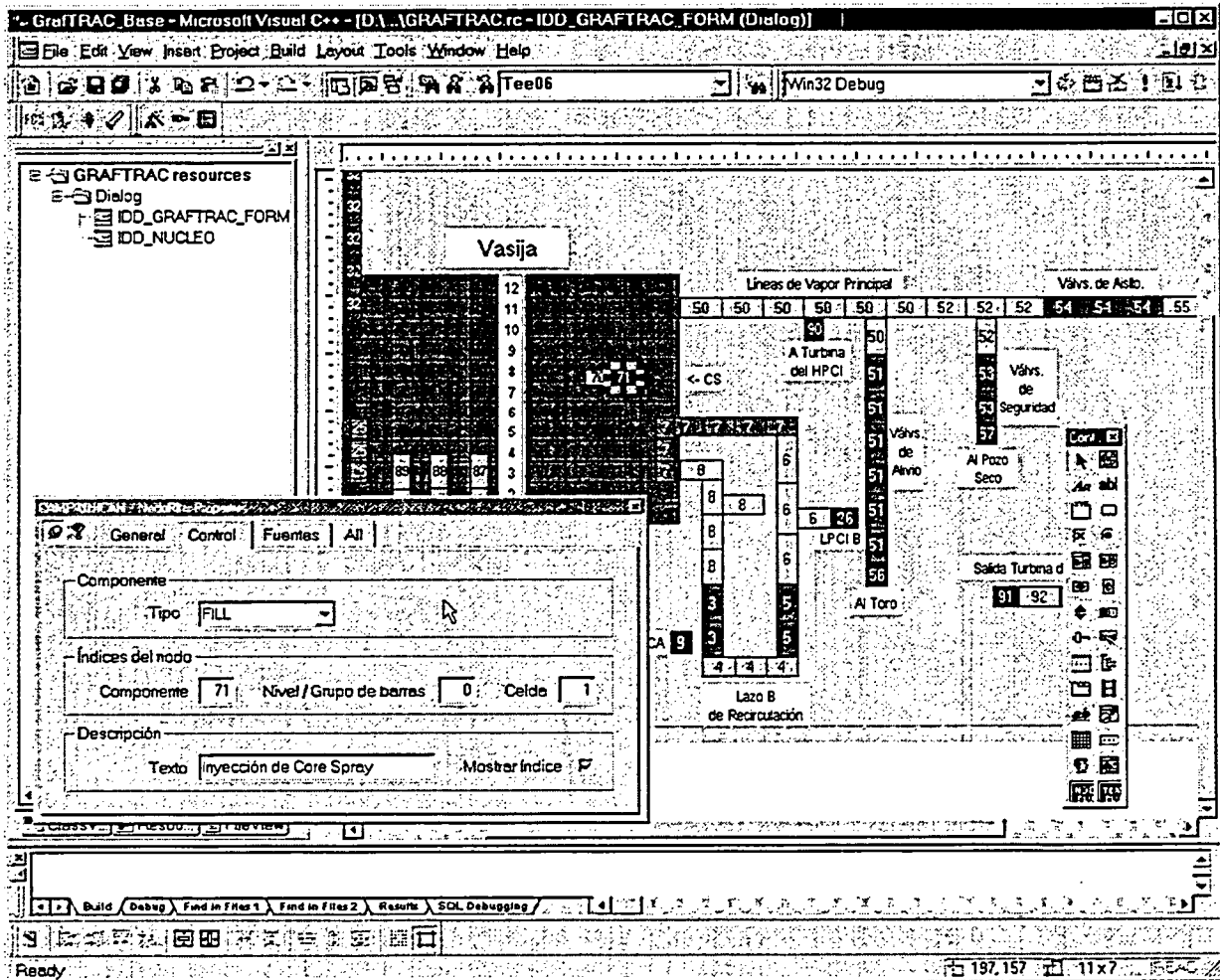


Figure 6. Working with Microsoft's Resource Editor on a copy of the pre-built project to modify GrafTRAC nodalization windows. Note that the controls toolbar contains the NodoPlus and TextoPlus icons.



## 2.6 Future Work

The main objectives of GrafTRAC development have been achieved and the application has demonstrated its capacity to be used as the only graphical post-processor for TRAC-BF1. Nevertheless, many improvements could be implemented in future versions. Some of them are mentioned next:

- Inclusion of data compression modules to reduce the disk space required to store the huge TRAC output files. There are some commercial libraries that would facilitate this process.
- Addition of more loading data modules able to retrieve different data formats used in the T/H simulation area. Data from other codes, as happens now with MAAP, could be easily compared to TRAC results.
- Development of an special module to create and modify the GrafTRAC interface without the necessity of using the Visual C++ environment. In a longer timeframe it would be desirable to accomplish the automatic generation of the interface from a TRAC input file without user intervention.
- The program could be translated to a multi-platform language like Java in order to generalize its use and execute it over the Internet. It must be pointed out that ActiveX technology has been specially designed for this kind of applications.

However, some of these features would be offered in a near future by the NRC's SNAP package for the consolidated TRAC-M code. We consider that, in the mean time, the present capabilities of GrafTRAC suffice to most of TRAC-BF1 users.

### 3 VisVas3D, a Dynamic and Interactive 3-D Visualization Prototype

VisVas3D has been intended as a prototype application to study the possibilities offered by three-dimensional visualization tools when working with complex simulation codes. We describe below the main features of the application in detail. In summary it can be defined as a dynamic and interactive visualization tool. On one hand, it offers an animated representation of the simulation results for the duration of the sequence being analyzed. On the other hand, the user is allowed to interact with the tool and modify many parameters of the visualization during execution: data refreshing rate, color palette, region of the model being represented, variable to visualize and so on.

TRAC-BF1/MOD1 was chosen as the target code taking into account the existence of a simple 3-D model, the VESSEL component, perfectly suited for our purpose. As happened with GrafTRAC, the application has been developed profiting from the object oriented programming paradigm and lets a third party tool cope with the harder tasks of graphics generation and rendering. Again, this third party tool is well known, easy to obtain and widely used. Regarding the platform of execution, VisVas3D is also, by now, a native Windows application.

As we show below, the application has a limited set of capabilities that could be extensively improved. However, it has demonstrated the ability of this kind of tool to provide code users with a new and more straight-forward insight within the great amount of data produced by simulations involving 3-D models. In the near future, a tool with very similar architecture to VisVas3D could be used to thoroughly analyze the results of much more complicated three-dimensional models.

### 3.1 Design Considerations

In order to get a valid point of view of the possibilities that these kind of tools could offer, we decided to develop an application that included all the capacities that could be required from them. The following were identified:

- Clear representation of data in a bi-dimensional screen. As we wanted to visualize a value that changes over time in a three-dimensional model we needed to represent snapshots of a function of three independent parameters. The 3-D model could be visualized by means of perspective on a flat window but the dependent parameter must be represented using a property of each point inside the virtual 3-D world. Obviously, the selected property was the color of the point according to a predefined palette.
- Those snapshots are taken each simulation time step and must be shown continuously to offer a complete visualization of the sequence. The application user must control this animated representation.
- Given the assumption that raw data would be obtained from a complex simulation code (at least, it includes a three-dimensional model) that could not be executed in real time on PC-platforms, the visualization would take place after the sequence has concluded. Thus, VisVas3D would be a graphical post-processor and should allow to visualize the simulation in real time, faster or slower depending on the user's choice.
- The user's vision of the 3-D world must be controlled by himself, being able to "move" freely around it while the sequence is being visualized.
- The structure of the 3-D model might imply that some parts of it remain hidden depending on the point of view selected by the user. So, the model representation should be, in some way, transparent and allow a certain vision of those hidden regions.
- The user must be able to modify the main parameters that define the visualization: points density, color palette, value ranges, refreshing rate, represented variable... and most of these changes should have an immediate effect on the visualization being executed.
- The application should provide as much detailed information as possible about the represented data: maximum and minimum values, value at a specified point and so on.

As in the GrafTRAC case, the general GUIs programming considerations also had to be taken into account. Finally, the tool should be easily distributed among the biggest possible amount of users so that we could obtain a valuable feedback from them.

### **3.2 Application Architecture**

The premises stated above and the necessity of obtaining the maximum benefit from the required programming effort, led us to develop a Windows based application using ActiveX technology. Again, to get a high efficiency during the development process and the best performance while running the application, we chose C++ as the programming language and Microsoft Visual C++ as the programming environment. Once more, we made extensive use of the MFC library. No need to explain the amount of resources required by an application of this kind in execution. However, we know well that last generation PC platforms are able to cope with it in such a way inconceivable a few years ago and, of course, they continue to be the most common platforms by far.

As explained before, we chose the VESSEL hydraulic component included in TRAC-BF1 as the three-dimensional target model. This component models a BWR vessel and its associated internals. The module uses a six-equation, two-fluid model to evaluate the flow through and around them. A regular cylindrical mesh, with variable mesh spacing in all three directions, encompasses the downcomer, core bypass and upper and lower plenums of the vessel. The model is complex enough to deserve a specialized 3-D visualization tool, but quite simple so it can be used in a prototype application as VisVas3D.

At the core of the program resides a third party ActiveX control: Avis2D, which is one of the software components of Digital Array Visualizer 1.0 [14]. This tool is distributed along with the professional version of Digital Visual Fortran 6.0, possibly the most common Fortran programming environment at the moment. It has been designed to let the user view and analyze array data graphically. Based on OpenGL, a high speed 3-D rendering library, it permits interactively move, rotate and zoom data graphs.

The Avis2D control provides more than 100 properties and methods that we have used to create, render and manage the images of our 3-D representation of TRAC data. Every time a simulated sequence is going to be visualized our application launches an instance of Array Viewer, the Array Visualizer application, which is responsible of handling all the graphics related tasks.

Previously, our program has loaded the relevant TRAC data from the graphic output file and defined an image of the vessel according to user's selections. It has to be pointed out that to avoid a solid figure that hides most of data, the vessel is represented by means of so many concentric cylindrical surfaces as radial rings are included in the nodalization of the component. Besides, the surfaces are created from a mesh of points with variable density. The color of each point in that grid depends on the value of the selected variable in the corresponding node, the value range and the active color palette. Straight lines colored by interpolation join the points. We have also programmed the response of the application to user interaction, the data refreshing management and the synchronization tasks.

Regarding the distribution requirements, a standard setup program for Windows platforms (Windows 95, 98 and NT) has been developed. It installs VisVas3D components along with the Avis2D control, Array Viewer and also a little TRCGRF file to be used as example.

### **3.3 Program Functionality**

In figure 7 we show a typical image of VisVas3D in execution. On the left side there is the main window: A bar-like dialog with buttons to start a visualization or close the program and also to enter the information required by the application before the visualization begins. In this prototype version, beside the name of the TRAC graphic output file (the "TRCGRF" binary file) and the parameters that define the image of the 3-D component (number of axial levels and angular separation between the points that conform the cylindrical surfaces), the user has to specify the nodalization of the vessel used in the TRAC case being analyzed:

radial rings, axial levels and angular segments with their respective values. The initial time interval for the animation of the visualized image also can be modified.

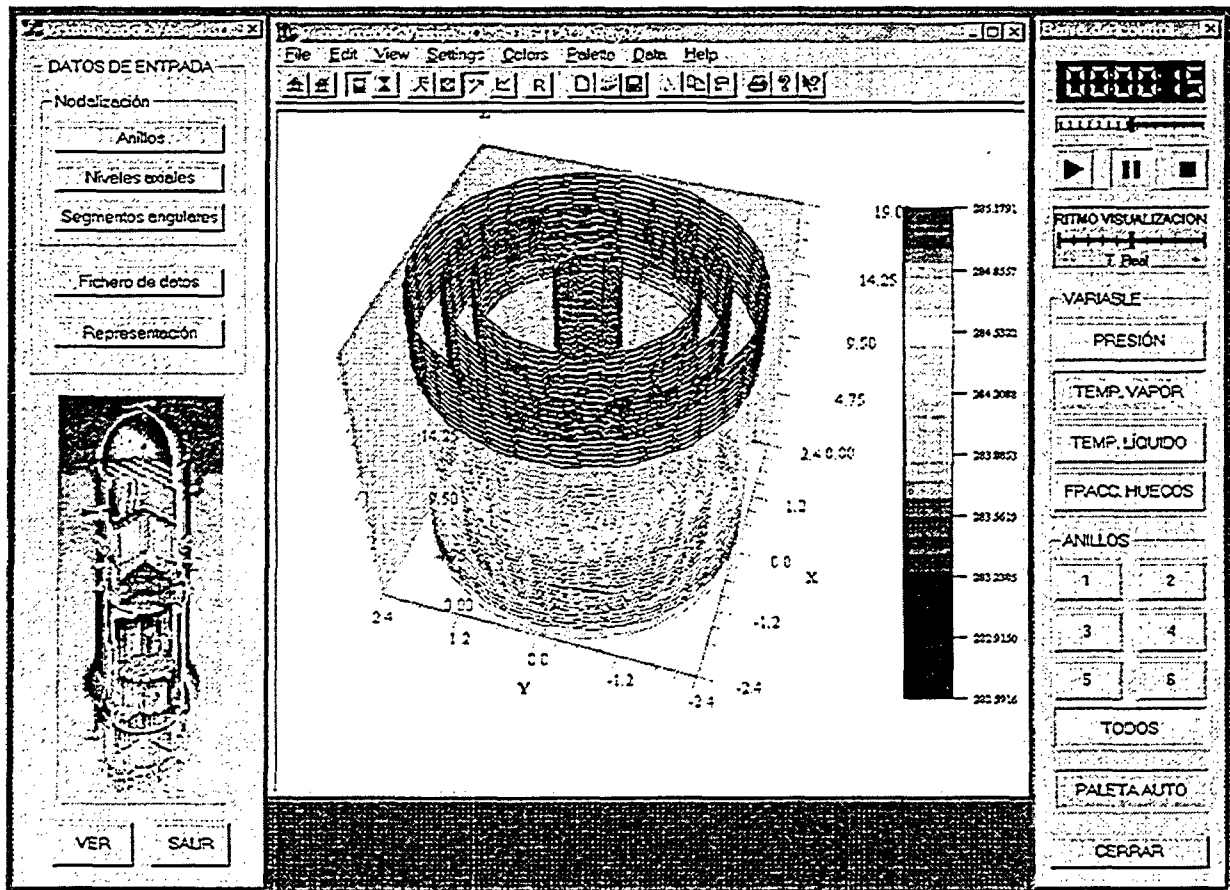


Figure 7. Typical display of VisVas3D showing the results of a TRAC-BF1 sequence.

When the corresponding button is pressed, Array Viewer is launched containing the representation of the vessel at the beginning of the simulation (central window in figure 7 with a four radial rings vessel defined by a mesh of 7400 points). On the right side appears a control bar to allow user's interaction while the visualization is running. On top of it a digital clock indicates the simulation time and a slide control shows the visualized time compared with the total simulated time. The latter receives input from the user and can be used to select the instant to be visualized moving forward or backward in the sequence evolution. Below, there is a set of "play", "pause" and "stop" buttons also available to control time and other

slide bar that allows to modify the data refreshing rate. The default value implies real time visualization but the user can slow down or accelerate the process in any time during execution.

By now, only fourth vessel variables can be visualized: vapor pressure and temperature, liquid temperature and void fraction. By means of a set of four buttons the user can select which one of these variables is being viewed. Another set of buttons enables the user to visualize only one of the vessel radial rings instead of the complete figure in order to obtain a clearer picture of the region of interest. Finally, there is a button to choose between a fixed color palette (adjusted at the beginning of the execution to the maximum and minimum values of each variable during the whole sequence) or an “auto” palette that changes every time step according to the range of values represented.

Moreover, the whole Array Viewer functionality is completely available. For example, it is possible to rotate the image in any direction or to zoom in or out by means of the mouse. If the cursor is placed in any point of the figure its four coordinates, i.e. point location and value of the selected variable at that point, are displayed. All the menu options offered by Array Viewer are also accessible.

### **3.4 Future Work**

As a prototype application, VisVas3D offers many areas of enhancement. Two of them are obvious. First, the possibility of visualizing any of the output variables of the VESSEL component, not only the four selected by now. Second, to include the capacity of interpreting completely the data output file so that the application itself finds out the vessel nodalization used in the simulated sequence without user's help.

In the long term it would be desirable to provide a more detailed information about the represented data. This goal could be quite easily achieved by means of a conventional visualization tool, like Xmgr5, that would be launched from VisVas3D to represent the time

evolution of the selected variable in any point of the 3-D model desired by the user. Other potential improvements include the ability to read the ASCII “ftn08” output file obtained when running the code in certain platforms and even the translation of the program to a platform-independent language like Java so it could be launched over the Internet. Finally, it would be very interesting to apply these techniques to a more complex 3-D model using, if needed, a different third party tool for graphics rendering.



## 4 Conclusions

Two new utilities for graphical post-processing of TRAC-BF1 results have been developed. Both take advantage of the latest advances on object oriented programming techniques and from the availability of powerful third party tools that could be easily accessed by their potential users. Moreover, the applications have been designed to run on Windows/PC platforms profiting from the continuous improvement in performance and cost reduction of this kind of equipment.

This approach has proved its efficiency more than enough. GrafTRAC, an integrated graphical postprocessor for TRAC-BF1, can be used as the only post-processing tool for that code since it incorporates all the functionality and the best features found in other packages designed with the same goal. VisVas3D, a prototype application for the visualization of the TRAC-BF1 Vessel component, has demonstrated the new insight on 3-D output data from T/H codes that specialized tools can offer.

## REFERENCES

- [1] J.L. Crespo, R.A. Fernández. Assessment of MSIV Full Closure for Santa Maria de Garoña Nuclear Power Plant Using TRAC-BF1 (G1J1). NUREG/IA-0122. U.S. Nuclear Regulatory Commission. 1993.
- [2] D.M. Snider et al. Nuclear Plant Analyzer Reference Manual Version 1.4. EGG-EAST-9096. Idaho National Engineering Laboratory. 1994.
- [3] K.R. Jones. Symbolic Nuclear Analysis Package. To be published on Proceedings of the OCDE/CSNI Workshop on Advanced Thermal-Hydraulic and Neutronic Codes: Current and Future Applications. OECD NEA. 2000.
- [4] SNAP Mod 0 / Version 0.6 User's Manual. U.S. Nuclear Regulatory Commission. 2000.
- [5] T.C Zhao, M. Overmars. Forms Library. A Graphical User Interface Toolkit for X. 1995.
- [6] T. Williams, C. Kelley. GNUPLOT: An Interactive Plotting Program. Dartmouth University. 1993.
- [7] DV-Draw and DV-Tools User's Guides. Dataviews Corporation. 1996.
- [8] P.J. Turner. ACE/gr Users Manual. Graphics for Exploratory Data Analysis. Oregon Graduate Institute of Technology. 1991-1993.
- [9] K. Jones. XMGR5 Extensions. Idaho National Engineering and Environmental Laboratory. 1998.
- [10] R.A. Earnshaw, N. Wiseman. An Introductory Guide to Scientific Visualization. Springer-Verlag. 1992.
- [11] G.M. Nielson. Scientific Visualization: Overviews, Methodologies and Techniques. IEEE Computer Society. 1997.
- [12] P. Corcuera, M.Garcés, J. Ryan. A Training Simulator with Soft Panels. Proceedings of the 2000 Western Multiconference. The Society for Computer Simulation. 2000.
- [13] Visual C++ User's Guide. In MSDN Library Visual Studio 6.0 Release. Microsoft. 1998.
- [14] Digital Array Visualizer 1.0 User's Manual. Digital Equipment Corporation. 1998.

**BIBLIOGRAPHIC DATA SHEET**

(See instructions on the reverse)

1. REPORT NUMBER  
(Assigned by NRC, Add Vol., Supp., Rev.,  
and Addendum Numbers, if any.)

NUREG/IA-0190

2. TITLE AND SUBTITLE

Nowadays Tools for Graphical Post-Processing of TRAC-BF1 Results

3. DATE REPORT PUBLISHED

MONTH YEAR

October 2000

4. FIN OR GRANT NUMBER

5. AUTHOR(S)

P. Corcuera, M. Garces

6. TYPE OF REPORT

Technical

7. PERIOD COVERED (Inclusive Dates)

9/2000 - present

8. PERFORMING ORGANIZATION - NAME AND ADDRESS (If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address; if contractor, provide name and mailing address.)

Applied Mathematics and Computer Science Department  
University of Cantabria  
SPAIN

9. SPONSORING ORGANIZATION - NAME AND ADDRESS (If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)

Division of Systems Analysis and Regulatory Effectiveness  
Office of Nuclear Regulatory Research  
U.S. Nuclear Regulatory Commission  
Washington, DC 20555-0001

10. SUPPLEMENTARY NOTES

11. ABSTRACT (200 words or less)

For the time being, there are no standard tools for graphical post-processing of TRAC-BF1 results. A new specialized application based on PC/Windows technology, GrafTrac, has been developed to integrate in a unique tool the best capabilities found in other post-processing packages. It simplifies the access to the output variable of interest by means of a very intuitive graphical interface and launches Xmgr5 to offer the whole functionality of this tool in treating data and plots. Besides, the user is allowed to customize the graphical interface with a simple procedure in order to implement a representation of the nodalization of the simulated model. A first release is being used by our group to analyze data from the last TRAC simulations. The users feelings are very encouraging. On the other hand, a clear tendency to improve T/H codes with the inclusion of detailed three dimensional models and by linking them to 3-D neutronic codes is observed. We have developed VisVas3D, a prototype application based on the simple TRAC-BF1 Vessel component, able to display 3-D information in a dynamic and interactive way. Running on the PC/Windows environment it serves to demonstrate the enormous possibilities that specialized tools can offer in this field. Th same programming approach has been used in the development of both projects: the latest advances in object oriented technologies combined with the utilization of well-known and easy to access third party tools to deal with the harder task of graphics generation and rendering. This solution has proved very efficient resulting in a fast development.

12. KEY WORDS/DESCRIPTORS (List words or phrases that will assist researchers in locating the report.)

TRAC-BF1  
Post-processing

13. AVAILABILITY STATEMENT

unlimited

14. SECURITY CLASSIFICATION

(This Page)

unclassified

(This Report)

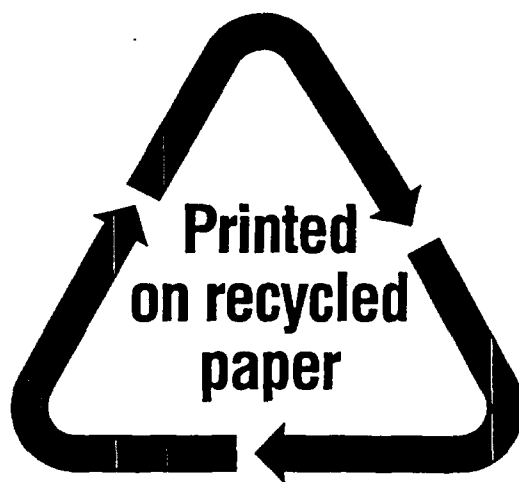
unclassified

15. NUMBER OF PAGES

16. PRICE







Federal Recycling Program



UNITED STATES  
NUCLEAR REGULATORY COMMISSION  
WASHINGTON, D.C. 20555-0001

