



DEVELOPMENT AND VALIDATION OF GUI BASED INPUT FILE GENERATION CODE FOR RELAP

M.M. ANWAR, A.A. KHAN, *I. R. CHUGHTAI, K.S. CHAUDRI, T. HAYAT¹ and M. H. INAYAT

Pakistan Institute of Engineering and Applied Sciences (PIEAS), P.O. Nilore, Islamabad, Pakistan

¹DOS, Pakistan Atomic Energy Commission, Islamabad, Pakistan

Reactor Excursion and Leak Analysis Program (RELAP) is a widely acceptable computer code for thermal hydraulics modeling of Nuclear Power Plants. It calculates thermal- hydraulic transients in water-cooled nuclear reactors by solving approximations to the one-dimensional, two-phase equations of hydraulics in an arbitrarily connected system of nodes. However, the preparation of input file and subsequent analysis of results in this code is a tedious task. The development of a Graphical User Interface (GUI) for preparation of the input file for RELAP-5 is done with the validation of GUI generated Input File. The GUI is developed in Microsoft Visual Studio using Visual C Sharp (C#) as programming language. The Nodalization diagram is drawn graphically and the program contains various component forms along with the starting data form, which are launched for properties assignment to generate Input File Cards serving as GUI for the user. The GUI is provided with Open / Save function to store and recall the Nodalization diagram along with Components' properties. The GUI generated Input File is validated for several case studies and individual component cards are compared with the originally required format. The generated Input File of RELAP is found consistent with the requirement of RELAP. The GUI provided a useful platform for simulating complex hydrodynamic problems efficiently with RELAP.

Keywords: Graphical user interface; RELAP, Thermal hydraulics

1. Introduction

The development of the RELAP family of codes was started more than 3 decades ago by the U.S. government. RELAP uses one-dimensional thermal-hydraulics, neutronics, generic and special components, control systems, and other models to describe the behavior of complex fluid-filled systems under single and two-phase conditions [1]. RELAP was originally developed to support the analysis of postulated accidents in commercial nuclear power plants in the United States. The code has been used to support basic research on two phase thermal-hydraulics, to design small and large scale thermal-hydraulic experimental facilities, research reactors, and commercial power plants; and to assess the safety of nuclear plants [2, 3].

Besides the aforementioned capabilities and application of RELAP, there are several problems associated with the writing of Input file. These problems are explained as under:

The input file of RELAP is a text file with “.”

extension. The total system flowsheet as well as the component properties, with all the initial conditions and operating conditions is described by specific card numbers [4]. The card numbers are generally six to eight numeric numbers. A single component may require several cards for complete definition of its parameters. These cards may sometimes be upto one hundred for a single component. User needs to memorize or has to browse throughout the manuals to write even a single card. This is a tedious job and time consuming as well. Moreover, if the solution is not converging, a lot of time and effort is required to modify this input file as the user is reluctant to browse the whole manual once again.

The main objective of Graphical User Interface (GUI) developed in the present work is to aid the RELAP user in preparation of input file [5]. The GUI has the following main features:

- The process flow diagram is drawn graphically so that the user can visually and interactively define the connectivity of various process

* Corresponding author : imran@pieas.edu.pk

components

- The physical properties and other parameters are assigned in graphical mode
- The GUI can also save and retrieve the process flow diagram with all relevant details for further processing and editing in a later session
- The default properties of each component are input automatically.
- Minor and major edits are done more easily and efficiently
- Large number of tables are entered more easily and in simple format
- The runtime help is provided for each component properties and
- Finally the input file from the process flow diagram alongwith component properties and the parameters is generated by just a single click.

Hence, use of RELAP becomes easier by this GUI and time and effort for input file generation and modification is much reduced.

A few of Graphical User Interfaces developed in the past lag in many ways especially in the saving and modification of entered properties [6]. The process flow diagram drawn and the properties once entered cannot be changed and user needs to work from the scratch. In this GUI such drawbacks are overcome and a more useful method is introduced.

2. Description of GUI

The GUI is developed in Microsoft Visual Studio .NET using Visual C Sharp (C #) as the programming language [7-9]. The GUI program is named as RELAP-GUI. In the program, user draws the Nodalization diagram graphically which is developed from process flow diagram to serve as a prerequisite for RELAP, assign physical properties, specify initial conditions and the other remaining input commands, and generate the input file of RELAP. The block diagram of our GUI is shown in Figure 1:

Brief description of various components of the program as shown in the layout is as under:

The GUI main form is the parent form. It contains title bar, menu bar, tool bar and drawing canvas. The component forms are called from this form. The graphical Nodalization diagram is drawn in the drawing canvas of main form using Nodalization diagram tools. When the Nodalization diagram is completed, the properties are entered in the respective component forms. A typical Graphical Nodalization Diagram in RELAP-GUI is shown in Figure 2.

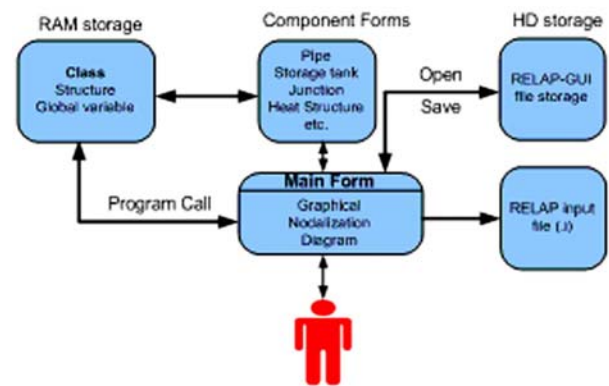


Figure 1. Block diagram of RELAP-GUI structure

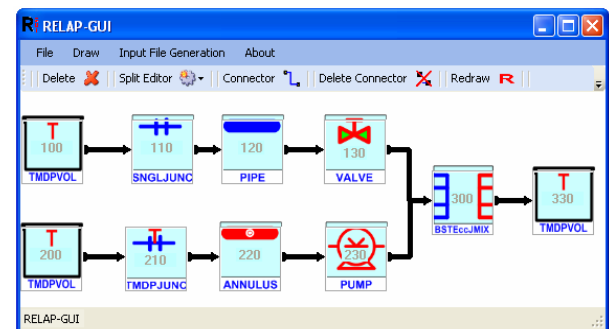


Figure 2: Graphical Nodalization Diagram in RELAP-GUI

The component forms contain all the properties and information needed for the input file generation. One of the component forms to input various properties required by RELAP for a typical pipe is shown in Figure 3.

The component forms added in RELAP-GUI are as under:

- Initialization Properties (including minor edits and plot requests)
- Hydrodynamic Components
- Single Volume (snglvol)

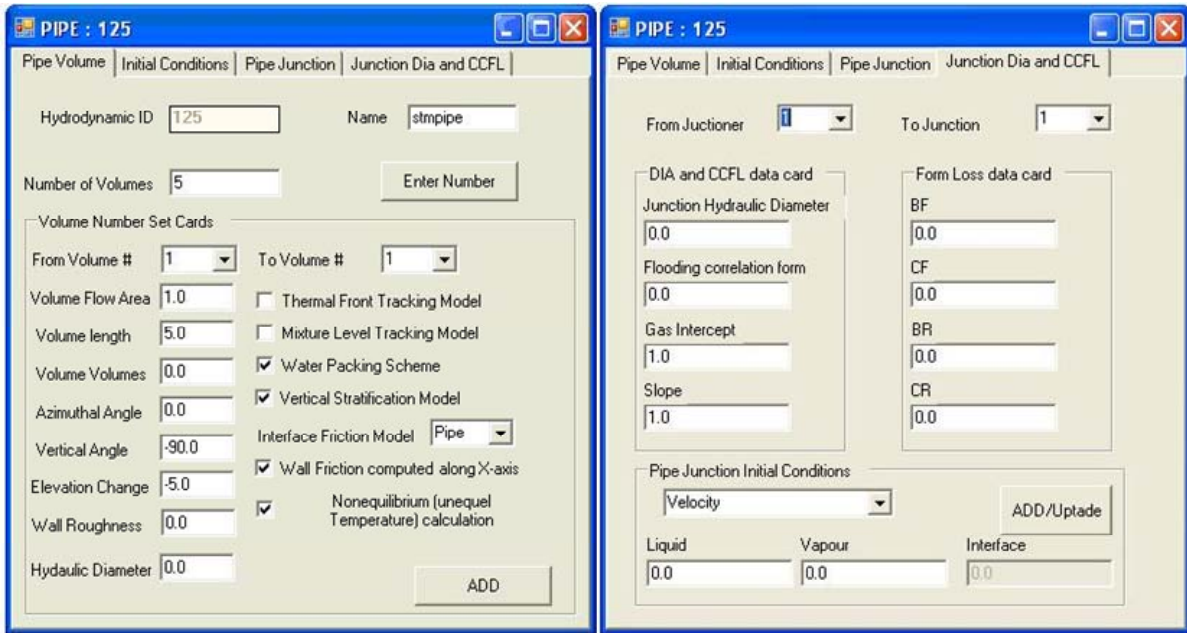


Figure 3. Properties Tabs of Pipe Component Form

- Time-Dependent Volume (tmdpvol)
- Single Junction (sngljun)
- Time-Dependent Junction (tmdpjun)
- Pipe (pipe)
- Annulus (annulus)
- Valve Junction (valve)
- Pump (pump)
- Branch, Separator, Jet-mixer, Turbine or ECC Mixer
- Accumulator (accum)
- Heat Structure Component

3. Case Study

A transient flow problem is taken from a RELAP tutorial. The problem is to simulate the flow from a high pressure tank to a low pressure tank via pipe. Two junctions connect the pipe with upstream and downstream pressure tanks. Water is the fluid medium with Air as a dissolved gas. The total flow through both junctions is required to be plotted by RELAP simulation code. The process flow diagram of the problem is shown in Figure 4.

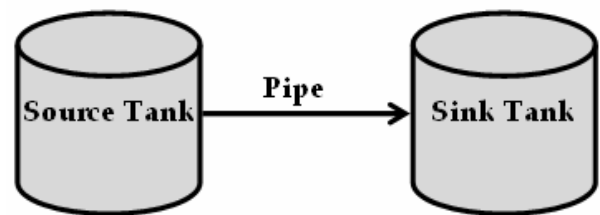


Figure 4. Process flow Diagram of problem

The Nodalization Diagram is shown in Figure 5.

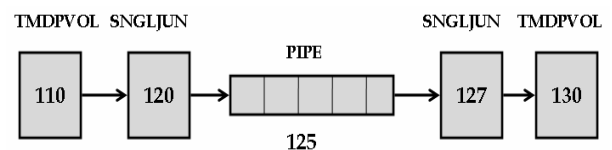


Figure 5. Nodalization Diagram of problem

The Graphical Nodalization Diagram in RELAP-GUI is shown in Figure 6. Some of the cards of the generated input file are shown in Figure 7.

The results of the problem simulated in RELAP using this RELAP-GUI generated input file are plotted in Figure 8

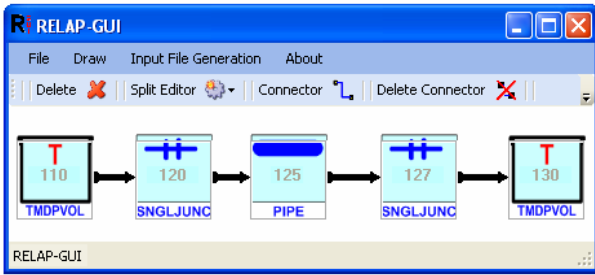


Figure 6. Graphical Nodalization Diagram in RELAP-GUI

```

=====
*          CODE GENERATED BY RELAP-GUI V-PRE RELEASE
*          COPY RIGHT @ PIEAS PAKISTAN
=====
= Sample Problem Input File Generated by RELAP-GUI
=====
*          PROBLEM TYPE AND OPTIONS card
=====
100 new transnt
101 run
102 british si
=====
*          Plot request
=====
20300021 p 125010000 1   ***Volume pressure
20300022 p 125030000 1   ***Volume pressure
=====
*          Hydrodynamic Component DATA
=====
*          Component Junction 120
=====
1200000 sngljuni sngljun
1200101 110010002 125010001 0.1 0.0 0.0 00000100 1.0 1.0 1.0
1200201 1 0.0 0.0 0.0
=====
*          Component PIPE 125
=====
1250000 stmpipe pipe
1250001 5
1250101 1.0 5
1251201 003 100.0 90.0 0.0 0.0 0.0 5
=====

```

Figure 7. Few cards of Input File generated by RELAP-GUI

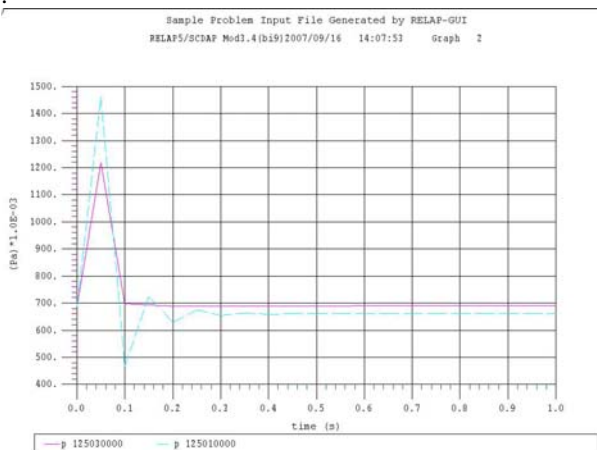


Figure 8. RELAP simulation results using RELAP-GUI generated input file

The results of the problem exactly match with that of the original input file provided with RELAP

software. All the components incorporated in the GUI have been verified by comparing individual component cards with the standard examples in RELAP5 Code Manual Volume V [2]

4. Conclusion

The Graphical User Interface for RELAP Input File Generation assist RELAP user in many ways. It is a much better approach to draw process flow diagram and assign properties of individual components. It is a fast and user friendly approach introduced with RELAP. The properties for input file generation can be saved and modified easily. The results of input file generated match with the manually written input file based on RELAP input file format.

References

- [1] RELAP/MOD Code Manual, Volume I: Code Structure, System Models, and Solution Methods, NUREG/CR-5535 (June 1995).
- [2] SCDAP / RELAP / MOD3.2 Code Manual, NUREG / CR / 6150, Rev.1 (June 1998).
- [3] E.A. Harvego, et al., "Developmental assessment of the SCDAP / RELAP / MOD3.2 code" 6th Inter. Conference on Nuclear Engineering, ICONE-6 (May 1998).
- [4] Appendix "RELAP-SCDAP 3.2".
- [5] R.R. Schultz, "RELAP5 Code Manual Volume V: User's Guidelines", Idaho National Engineering and Environmental Laboratory Bechtel BWXT Idaho, LLC.
- [6] G. Mesina and J. Galbraith, "RGUI 1.0, New Graphical User Interface for RELAP5-3d", 7th International Conference on Nuclear Engineering Tokyo, Japan, April 19-23 (1999).
- [7] L. Serban, D. Brezoi, T. Radu and A. Ward, "GDI+ Custom Controls with Visual C # 2005", PACKT Publishing.
- [8] J. Liberty, "Programming C #", Third Edition, O'Reilly & Associates, Inc.
- [9] S. Robinson, C. Nagel, J. Glynn, M. Skinner, K. Watson and B. Evjen, "Professional C #", Third edition, Wiley Publishing, Inc.