IMPLEMENTATION OF RUNTIME VISUALIZATION FOR TOUGH2

H. Xin Yang
CRWMS M&O/INTERA Inc.
101 Convention Center Drive, P-110
Las Vegas, NV 89109
(702) 295-4692

Srikanta Mishra
CRWMS M&O/INTERA Inc.
101 Convention Center Drive, P-110
Las Vegas, NV 89109
(702) 295-4655

ABSTRACT

This paper is a summary of the work conducted for the TOUGH2 runtime visualization in support of the modeling of the unsaturated flow regime at Yucca Mountain.

I. INTRODUCTION

TOUGH2 is one of the numerical simulation programs used in the modeling of unsaturated zone fluid flow and heat transfer for the potential nuclear waste repository at Yucca Mountain. TOUGH2 offers a wide range of capabilities and features, including the flexibilities to handle different fluid mixtures and facilities for processing geometric data, but none in the area of data output visualization. This paper presents the work that has been undertaken in the implementation of runtime visualization of the output generated by TOUGH2.

II. DESIGN AND IMPLEMENTATION

The first step in runtime visualization is to derive an efficient data structure to represent the object to be modeled. In TOUGH2 the entire irregular shaped data domain is constructed from a collection of data objects. To accommodate grid blocks and future enhancements, a general data format was chosen to represent the grid geometry. The data file format (shown in Fig. 1) that has been chosen to represent this data object is based on the most commonly used Object File Format, with the enhancements that the file also contains the names of the data elements.

Following the foundation work for structuring the data, the graphics section was implemented. TOUGH2 is written entirely in Fortran, but the most effective way of implementing computer graphics on Unix workstations is using C/C++. In order to simplify the calling and linking of the graphics module with the various TOUGH2 modules, the entire C functions which call the X windows libraries are encapsulated through following functions: *initGraph()*, *closeGraph()*, *updateData()* and *showGraph()*.

As shown in Fig 2, *initGraph()* is called at the very beginning of a TOUGH2 simulation. This function allocates the necessary memory blocks for data display, then the plain geometry display of the model is shown on screen. During each simulation, TOUGH2 calls *updateData()* and *showGraph()* which update the display of scalor or vector data fields depending on the specification parameters. In addition, the display for each simulation is captured and stored into a sequence of .GIF files. This further enhances the ability of the analyst to review the execution sequence. At the completion of the simulation, the *closeGraph()* is called to terminate the connection to the display window and to free the memory blocks used for data manipulation.

III. SUMMARY

This runtime visualization design for TOUGH2 been successfully implemented. This paper will present a real-time demonstration of the various visualization options implemented to date.

IV. REFERENCES

1. Nye. Adrian, *Xlib Programming Manual.* O'Reilly & Associates, Inc.

2. Nye. Adrian and O'Reilly Tim, *X Toolkit Intrinsics Programming Manual.* O'Reilly & Associates, Inc.

Module Name	Data Description
Header Line	Number of elements and nodes with optional data range specification
Vertices	Coordinates x, y, z of each specified node
Name/Connections	Name referenced in the TOUGH2 code, followed by a counter specifying number of nodes involved then the node IDs.

Figure 1. Outline of the data structure used to represent irregular geometry

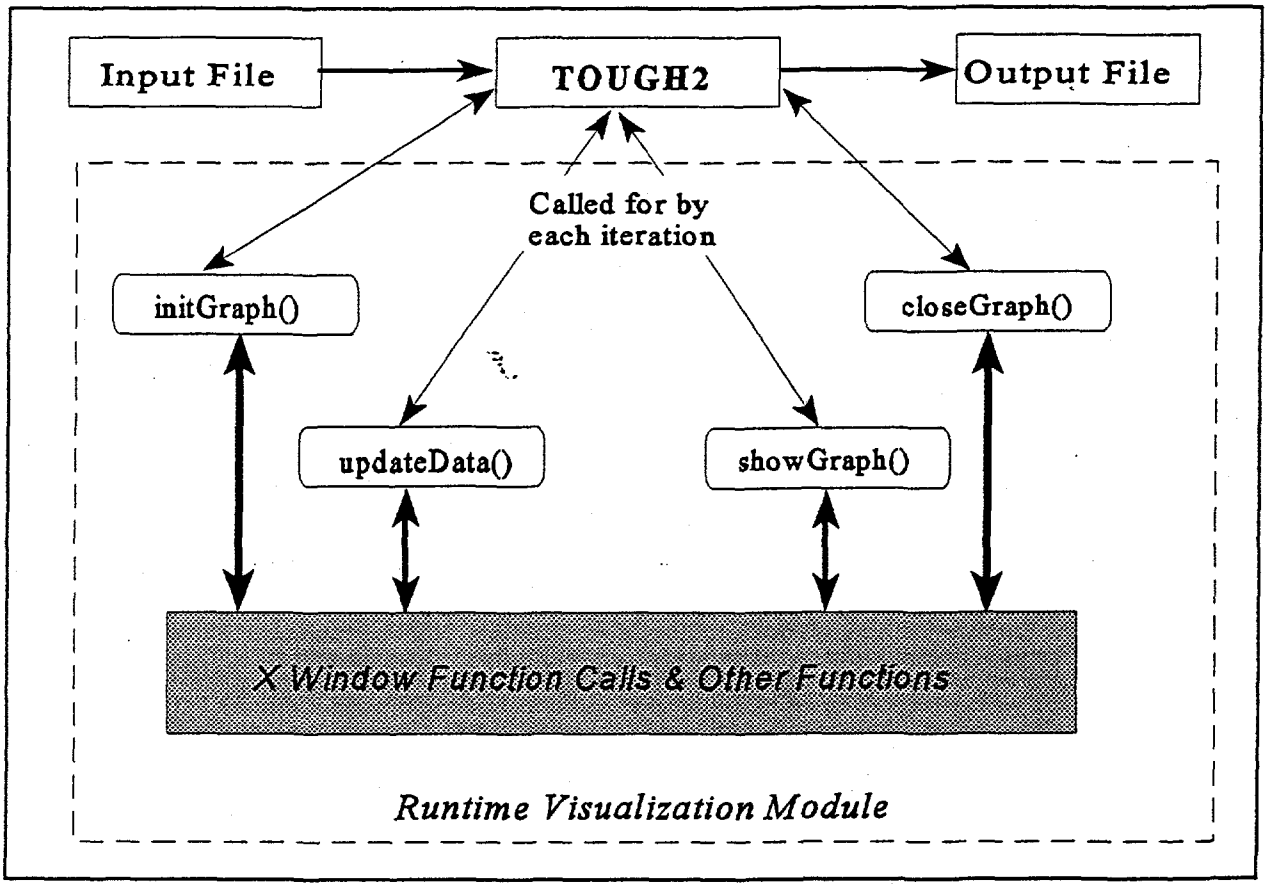

Figure 2. Program flow for TOUGH2 runtime visualization

DISCLAIMER