

ONLINE MODELING IN TEACHING GEOMECHANICS VIA THE WEB SITE

Jiang Li and Matthew Y. Lee
Morgan State University/Amherst College

Abstract

In the present paper, the design of online modeling in soil mechanics with multimedia tool on the Internet has been discussed. The online modeling in geomechanics servers as a virtual laboratory that can be used for both teaching and research at the Department of Civil Engineering, School of Engineering, Morgan State University. In this paper, the follows are emphasized: 1) design of main and sub web pages, 2) design of the online run time ActiveX coded with Visual Basic 5.0 for computation and graph, 3) the layout of online interface through WebPages and the online application for modeling. This is the first attempt to combine applications of Visual Basic to the web page on the Internet or the Intranet with teaching and research in conceptual, physical, mechanical and mathematical modeling in soil mechanics using one-dimensional consolidation theory so that problems in engineering such as land subsidence due to ASR (Aquifer Storage and Recovery) applications can be simulated and predicted. Therefore, this online modeling tool not only provides an alternative way to pass knowledge and to enhance students understanding of principles taught in geomechanics, but also it offers an online useful and innovative laboratory with more flexibility and convenience to help students in learning.

I. Introduction

Thanks to the ever-increasing World Wide Web, in both size and popularity, one may circle the globe in the comfort of his or her own favorite chair. It has allowed instantaneous interaction between people from all corners of the world. This mode of instantaneous communication is unparalleled in all communications technology. We should exploit the powerful potential of the Internet by implementing a utility through which the urbanized Internet can be transformed into a powerful and flexible instrument for academia (1 and 2). The accuracy of digital data in conjunction with the bettering speed of Internet connectivity, the Internet has become an excellent candidate for scholastic usage. Currently, there are technologies that allow students to “attend” courses and lectures through the marvel of the Internet and Intranets (3, 4 and 5). Although the current development of applications using the multimedia tools provides convenience to the students, it does not provide an interactive media through which the students may actively learn. The present paper presents one example of the solution to this form of problems. It is an interactive method with which students in engineering or in soil

mechanics are able to participate in online labs. Although this is not as media rich, it provides the key ingredient to successful learning that is currently missing: truly interactive presentation of course related materials. This paper will explore the creation process and the practical applications of the online labs in addition to the future direction and plans with this tremendously helpful technology in the advancement of student's education. Multitudes of other current as well as developing technologies will be briefly examined for their relevance and potential applications in this innovative mode of learning.

As computer hardware continue to dutifully shatter previous limits, they become an integral component in developing better visualizations of the complex models and mathematical theories used in engineering. Software utilized in this area of development is as equally available as hardware with its ability and complexity compounding just as rapidly. Many engineers are familiar with the older languages such as FORTRAN, PASCAL, C, etc. used frequently in the area of mathematics and engineering sciences. The pitfall of these legacy computer languages is that they provide highly limited or no graphical and three-dimensional displays to visual and user-friendly interfaces that are essential to application developments in software engineering. With the production of newer programming interfaces and languages (e.g., Microsoft's Visual Studio), it is now possible to overcome these previous hurdles to bring forth better representations. The selection process of a proper language from the plethora of available possibilities needs to take three major considerations. First, a selected language must contain the solutions necessary to the project of representing complex models and outcomes. This is, of course, the main priority in choosing a language that best fits the product. Second, the language itself must not present a barrier. The language's intricacy and easy of use must be carefully balanced for optimal performance and time efficiency. Third, support for the language must be adequate. This problem presents it self in situations where a language has fulfilled the first requirement but is not very popular and is therefore not supported widely on the network, thus inhibiting proper performance.

II. Design of Web Pages for the Online Modeling

A web site is designed for the analytical computation and modeling for one-dimensional deformation of aquifer systems with consolidation theory in soil mechanics. The WebPages at this site are designed or organized by four levels (see Chart 1). The first level has one main web page, the second level contains one sub-page, and the third level has four sub-pages that include two sub-pages respectively for the detailed computation at the forth level. The following will discuss the design of WebPages at each of these levels.

At the first level, it is a welcome page with the brief information and the table of content that can be navigated with drop-down note buttons. The navigating bar is also designed along with all other pages for the user's convenience. At the second level, on this page illustration of the task and goal of the online laboratory is given. At the third level, there are four components. They are *Introduction*, *Modeling with Sinusoidal B.C.* (Boundary Condition), *Modeling with Trapezoid B.C.* and *Modeling with Triangular B.C.* Each sub-

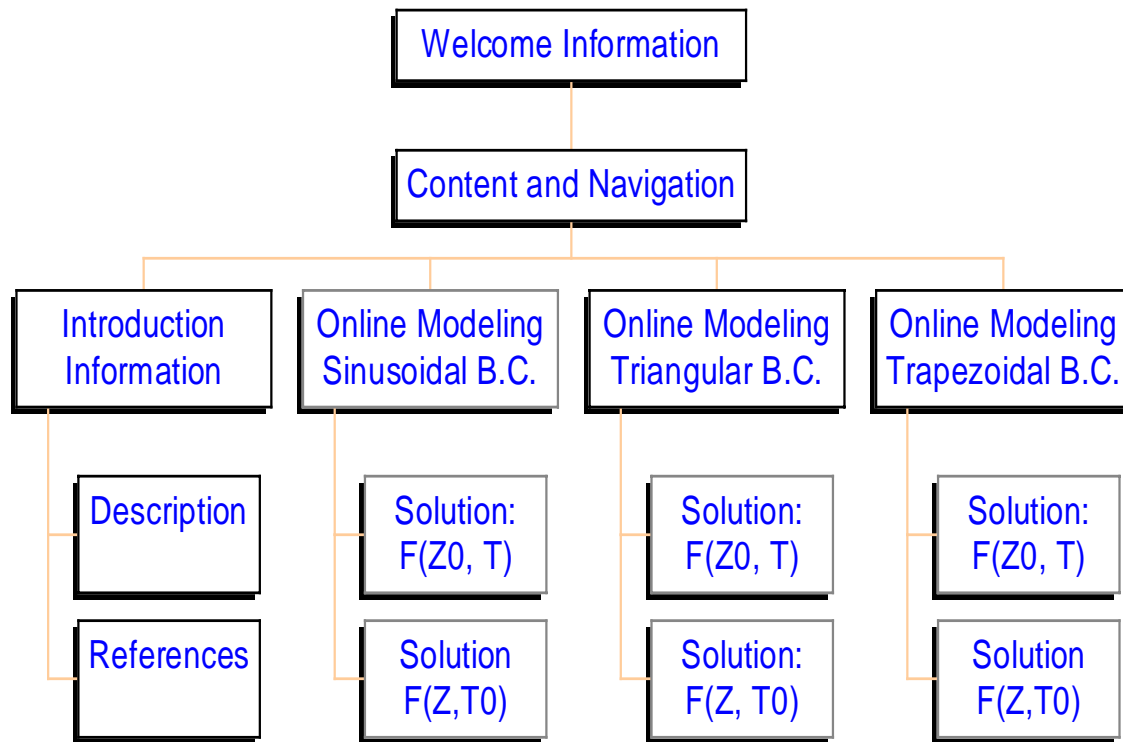


Chart 1. The Flow Chart of Design for WebPages

page at this level contains other two sub-pages at the sub-level. At the last level, there are total eight sub-pages that are the core of the online laboratory and play a key role in the online computation (Figure 1 demonstrates the appearance of one of the eight pages).

Introduction includes *Description* and *Reference*. *Description of the Problem and Significance* states the related problems in engineering, the significance to solve this problem. On this page, the literature review for past research and current investigations is included as well. *Reference* lists the selected research papers that contain the related information such as analytical solutions with specific boundary conditions and discussions of special cases of the solution. It is important for one to read this information before to use the online modeling tool.

Modeling with different boundary conditions has similar components (e.g., *Modeling with Sinusoidal B.C.*, *Modeling with Trapezoidal B.C.* and *Modeling with Triangular B.C.*). In each web page, the following information is included (see Figure 1):

1. A conceptual model shown by a diagram with a sandwich model that has the sand-clay-sand pattern with the geometric and pumping information,

2. A mathematical mode is illustrated by a diagram with the complete mathematical description including the governing equation and boundary conditions.
3. Boundary conditions show changes in effective stress at the boundary between the sand and clay layers. The effective changes are induced by the pumping-injecting activities.
4. Analytical solution and results from computation is based on the mathematical model or description, and is shown with the result curves in two diagrams on each page.

Online Application Program coded with Visual Basic 5.0, one component of Visual Studio (one of Microsoft application packages, plays a core role in the online analytical computation and molding. The next section will discuss the design for this application.

III. Application Design and Programming with Visual Basic

The interface for the Visual Basic program is intended to be straightforward and basic. Application design includes two parts. One is application layout on a main form and the other is the program code design with Visual Basic 5.0. The Visual Basic is easy to learn, and its applications are convenient to use.

The application layout on the main form comprises: a parameter input field, a command input field, and a menu bar (this is only available on an offline version). In the parameter input field, input parameters are labeled with its respective text box. First, the parameter input field is design for convenience for users who run the program and adjust calculating parameters. The parameter input field contains 13 parameter input boxes that allow the user to supply the parameters needed and described in the mathematical model for each of the three boundary conditions. Some default values are provided for the inputs but may be changed at the user's needs. Second, once all the necessary input is in place, the user may then execute the mathematical model in the command input field in which three important function keys are provided: **Run**, **Save**, and **Plot** for executing the program, saving the calculated results and plotting the solution curves. Third, the menu bar not only provides the *File* items that duplicate the function as the command input field does, but also provide the *Edit* and *Help* menus for user's conveniences. The menu bar, however, only works on an offline version since the online ActiveX control is not allotted a drop-down menu as in normal applications.

The design of the code for the application includes: computation, graph plot, result output and progress indicator. The computation is trigged by clicking on the **Run** button that is present on the interface. The core code for computation of the analytical solution from the partial differential equation with specific boundary conditions is programmed with VB 5.0. The results are stored in a designed array for plotting the graph later. At the same time, a text file of computation data can be formed and is ready to output if one clicks the **Save** button. The function will save the current data from memory into a formatted text file chosen by the user. This data file can also be further imported to a spreadsheet program such as MS Excel for detailed graphing. A coded function for plotting graph of solution is called when the user wants to plot the curve by clicking the

Plot button. A progress indicator located near the bottom of the main interface is embedded in the program to show the process of the running program.

The graphing of the data occurs at the user's demand. It is a step that is used only after the calculation command has been completed. The data from the current calculation remains resident in RAM until the next command to execute a calculation when the new results will overwrite the previous set of results. When the draw command is access via the button on the front panel or the menu, the current set of data is sent to the Microsoft Chart ActiveX control (i.e., the mschart.ocx control). From there, the graph is displayed based on the data from calculations derived from the given parameters provided on the front panel.

A more important aspect of this program is behind the interface and inside the code. The program was developed with Visual Basic 5, a part of Microsoft Visual Studio. Visual Basic is largely an IDE (Integrated Development Environment) within which a programmer may stitch together portions of code that control an array of ActiveX controls from Microsoft and many other third party vendors. For this program only Microsoft ActiveX controls were needed and used (i.e., comctl32.ocx, comctl232.ocx, mschart.ocx, comdlg32.ocx). From the controls used, many are simply components of the main interface. MSCHART.OCX is the control named in the previous paragraph that is responsible for the graphing capability. The reason Visual Basic was chosen in the plethora of development languages and tools is that it is based almost entirely on ActiveX controls. This makes porting the application to the Internet extremely easy. As a matter of fact, Visual Basic 5 includes a utility called the Application Setup Wizard that guides the developer through a series of steps to arrive at an independent and Internet capable final product. Various code subroutines direct the actions of the ActiveX controls and feed data to the graph control. The code responsible for the actions of the interface and the code responsible for the calculations are separate. Individual functions are devoted to the calculation and summation that are described clearly in the respective mathematical models.

IV. Summary and conclusion

In conclusion, there is much to be gained through an innovative program such as the one described in the current paper. This is the first attempt to combine the web page design on the Internet with applications of Visual Basic 5.0 to establish an online modeling tool for teaching and research in geomechanics. Mainly, this stems from the saving of resources that are the usual limits on any educational institution. These resources are not only fiscal but also abstract yet important entities such as time. One of the most obvious limitations on resource is the physical one. Most likely, computers, classroom space, and class hours are very valuable to both the faculty as well as the students. In order to maximize and optimize such times, many important educational materials are left out due to the insufficient availability of these finite physical resources. Since this program can be posted on the Internet, it increases the available time and the accessibility of the online lab. Thus, the opportunity for learning is widened. Similar to the physical boundaries are the software limitations. These mostly spring from financial restrictions with regard to

the purchases and usage of specialized software packages. Software as well as software licenses are getting progressively more costly and complex and thus have become a category of obstructions of their own. With “home-made” applications such as these, the need for software licenses may be abolished or replaced with a simple EULA (End-User Licensing Agreement). Tight fitting schedules are also closely associated with the physical constraints. It is usually difficult to select a lab hour that will well suite everyone. This is easily removed since the online lab will be located on the Internet and accessible at all hours. In solving these current problems, such online labs will offer many advantages that will make learning more efficient and effective for both faculty and students.

Bibliography

1. Wilson, J. M. & Byron P. R. (1996). A multimedia model for Undergraduate Education, *Technology in Science*, Vol., pp. 315-387.
2. Holzer, S. M. & Andruet, R. H. (1998). Learning statics with multimedia and other tools, ASEE Annual Conference and Exposition, Seattle Washington, pp120-126.
3. Li, J. and Lee, Y. M. (1998). Teaching Mechanics with Multimedia Tools, the proceedings of the 1999 ASEE annual conference at Charlotte, NC. USA, June 20-24, 1998, CD-ROM, Session No. 1668.
4. Li, J. (1998). Teaching Engineering Mechanics with the Internet, Abstracts in Proceedings of Workshop of Reform of Undergraduate Mechanics Education, Pennsylvania State University, August 16-18, 1998.
5. Li, J. and Lee Y. M. (1998). An Experiment of Teaching Engineering Courses with the Internet, in Proceedings of the 1998 Fall Regional Conference of Middle Atlantic Section of ASEE for Engineering Education, Howard University, Washington D.C., pp. 78-83.

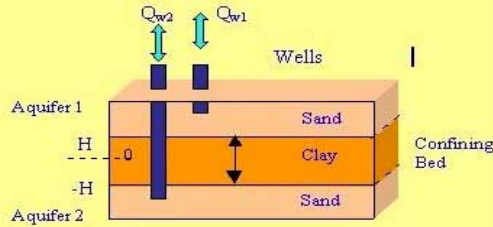
JIANG LI

Jiang Li is an Assistant Professor at the Department of Civil Engineering, School of Engineering, Morgan State University. He received his BS degree in Electronics Engineering from Wuhan Poly-technology University in 1982, MS degree in Civil Engineering from Chinese Academy of Sciences in 1989, PhD. degree in Hydrology and Hydrogeology from University Nevada-Reno in 1994. He is a member of ASEE, ASCE, AEG, AGU, NGWA, NYAS, AAUP, etc

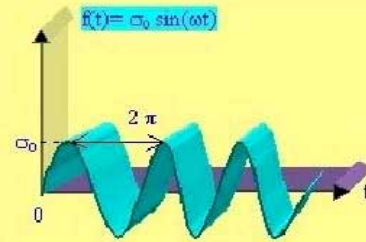
MATTHEW YU LEE

Matthew Yu Lee is a junior college student at Amherst College, Amherst, MA. He is currently pursuing his BA degree with a double major in Computer Sciences and Chemical Engineering.

Numerical Modeling for One Dimensional Subsidence with a Sinusoidal Boundary Condition (Z, T0)



The conceptual sandwich model



The boundary condition

$$\frac{\partial u}{\partial t} - c_v \frac{\partial^2 u}{\partial z^2} = B \quad (1)$$

$$u(z, 0) = 0, \quad -H \leq z \leq H \quad (2)$$

$$\frac{\partial u}{\partial z} = [a_1 t / E_{1c} + \sigma_{01} f_1(t) / E_{1v}] \quad z = H \quad (3a)$$

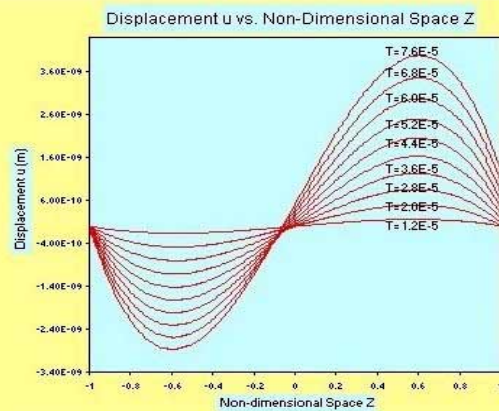
$$\frac{\partial u}{\partial z} = [a_2 t / E_{2c} + \sigma_{02} f_2(t) / E_{2v}] \quad z = -H \quad (3b)$$

The mathematical model

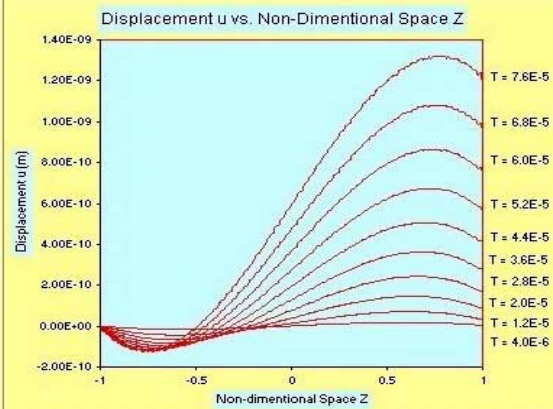
Click Below to Run Program

The program is coded for one dimensional subsidence

Run Program



The output results for $a_1 = a_2$



The output results for $a_1 > a_2$

Figure 1. The Appearance of One of Eight Sub-pages Designed for Online Modeling

Analytic Solution of 1D Deformation of the Clay Layer Sinusoildal B.C. (Z0, T)

By Dr. Jiang Li

Input Parameters

E at Boundary=

H (1/2 Thickness)=

T0=

B(Constant)=

a1(Upper B.C.)=

z=

a2(Lower B.C.)=

Real time t=

p1(Upper B.C.)=

Cv =

p2(Lower B.C.)=

f1(Upper B.C.)=

f2(Lower B.C.)=

Progress:

Figure 2. Appearance of User-friendly Layouts for the Online Computation