

AC 1996-504: Using Java to Develop Interactive Learning Material for the World-Wide Web

Benjamin "Quincy" Cabell VI,

Javed Alam, Youngstown State University

Joseph Rencis, University of Arkansas

Using Java to Develop Interactive Learning Material for the World-Wide Web

Benjamin “Quincy” Cabell V¹ and Joseph J. Rencis²
 Mechanical Engineering Department
 Worcester Polytechnic Institute
 Worcester, MA 01609-2280

Javed Alam³
 Department of Civil and Environmental Engineering
 Youngstown State University
 Youngstown, OH 44555

ABSTRACT

Java has recently emerged as a powerful programming language for developing platform-independent, interactive and computational based software that can be used on the World-Wide Web through a Java-enabled Web browser. The paper introduces the Java programming language, its advantages and disadvantages, and its characteristics for developing interactive instructional applications on the World-Wide Web. The interactive and computational capabilities of Java are demonstrated through a simple matrix assembly applet (a piece of networked software). This applet allows the student to assemble element equations into the global (assemblage) equations for the finite element method (FEM). The matrix assembly applet features a graphical user-friendly interface, on-line help and interactive feedback.

The authors are currently using Java to develop a prototype interactive learning tool for the one-dimensional bar element. The interactive learning tool for finite element method is called FEMur-CAL (Finite Element Method universal resource Computer-Assisted Learning). This FEM tool does not replace the conventional classroom experiences, but provides supplementary instruction to students who need extra help. The prototype will be integrated into the ‘Learn the Finite Element Method’ component of the Finite Element Method universal resource (FEMur). FEMur is a World-Wide Web site maintained by the authors of this paper and its Internet address is <http://amber.wpi.edu/~FEMur>.

1. INTRODUCTION

The development of the Internet, a network of networks, allows the interconnection of computers located at distant locations. The number of users and Internet hosts are growing at an exponential rate and there does not appear to be an end in sight. The Internet offers a great potential for the rapid, cost effective exchange of massive amounts of information. The introduction of hypertext transport protocol (HTTP) used in the current World-Wide Web (WWW, or the Web [1, 2]) project allows the FEMur development team to create information material in hypermedia format. It is a combination of Hypertext [3, 4] and multimedia. It allows one to combine information from such diverse sources as plain text, pictures, sound clips and animation/video clips to-create hypermedia documents using the Hypertext Markup Language (HTML). These documents are placed on a computer which acts as a WWW server allowing easy and instant access worldwide by anyone with access to a computer with Internet access and a WWW browser (such as Netscape/Mosaic [5, 6]).

¹ Undergraduate Student <thethe@wpi.edu, <http://amber.wpi.edu/~thethe>>

² Associate Professor <jjrencis@wpi.edu, <http://amber.wpi.edu/~jjrencis>>

³ Professor <fm010201@ysub.yosu.edu, <http://jove4.eng.yosu.edu/civil/javed.html>>



A significant part of the teaching process involves the delivery of course related information to the students. The recent developments in computer networking technology offer a new way to perform this function. There have been several attempts to use the development of the WWW for instructional purposes. These efforts are directed towards creating course material in multimedia format using HTML [7-9] and making it accessible through the Internet. The student can access this material from remote locations. The student's role has been confined to passively browsing through the information available on the Web pages. This use of WWW for teaching and learning lacks the interactive component essential for the learning process.

The HTML language is continuously evolving to incorporate more features needed to produce multimedia documents. In its current form, Web browsers use helper applications to display files which they do not internally know how to display (such as Microsoft Word documents, Post Script files, and audio/video files). The recent introduction of the Java programming language [10] extends the capability of HTML documents by including requests for Java applets in the source documents. This process allows developers to integrate functions formerly available only through externally launched helper applications. Java provides the freedom to create interactive content for the Web by developing new data types and the methods to operate on them.

When a Java enabled browser (such as Netscape 2.0) encounters the proposed HTML applet tag, the browser request for the Java class (code) specified by the applet tag; the browser iteratively loads each class needed by the classes already loaded until all necessary classes are downloaded. Java classes are code segments (in compiled form); each class encapsulates data and the functions necessary to manipulate that data. These classes are generally stored on the same machine as the HTML documents which reference them, though they could be stored anywhere. When the requested classes are received, the Java enabled browser executes the code. The beauty of Java is that there are no specific limitations on what the Java language can be used to do. For security reasons, however, there are restrictions on how Java code can access the computer on which it is run (in this way, viral and Trojan horse dangers are averted).

Java applets can display graphics (still or animated), play audio files, and receive user input from the mouse and keyboard. Anything that can be done currently through familiar operating systems such as MS Windows and MacOS can be done with Java. Web documents come alive because the computer can respond instantly to a user's input; no longer is the user forced to supply the computer with responses via fill-out forms submitted through common gateway interface (CGI) requests. The punch cards of WWW interaction have been replaced with magnetic tape. The computer can now become an active participant in the information dissemination process, rather, than a clumsy cathode-ray book. The computer platform independence of Java language makes it an ideal programming language to develop interactive and computational based applications for engineering education on the WWW.

2. OVERVIEW OF JAVA

Java is both a program environment and a language developed by Sun Microsystems, Inc. [10]. Java has been tailored specifically for networked computing, such as that which takes place on the Internet. Languages such as C++, Pascal, and FORTRAN require extensive libraries and experienced programmers to achieve the same degree of networked functionality. Any program that is written in Java can run on any computer as long as the "Java Virtual Machine" is inside. The Java Virtual Machine is real standard for an imaginary machine. The standard reflects the basic capability of all computer platforms (all CPUs as well as all operating systems). With Java, a programmer no longer writes code for a particular real platform but instead for this imaginary computer.



Each real computer can run this platform independent code by becoming (through software) this Java Virtual Machine.

Java is an object-oriented programming (OOP) language based heavily on C++. The Java programming language allows the developer to build both **applets** and stand-alone applications. **Applets** are pieces of compiled code that are referenced within Web pages (HTML documents) on the WWW. An **applet** requires a Java-enabled browser such as Netscape Navigator 2.0 [5] to run. An example of an **applet** is the matrix assembly program in Section 5. Stand-alone programs are written in almost the same way as **applets**, but do not require, nor do they use, a Web browser; almost all computer programs in common experience are examples of stand-alone applications: Microsoft Word, Corel Draw!, and Lotus Notes.

The Java source code is written in the same way as code for any other language. New tools and integrated development environments (IDE) are becoming available all the time, for MS Windows 95, MS Windows NT, Macintosh MacOS, Sun Spare Solaris, and soon for any Java capable machine. Java source code is compiled using a Java compiler, which creates a binary bytecode format that is platform-independent. The bytecode is unlike normal machine code in that it is written for the Java Virtual Machine and is not, therefore, native to any particular processor or operating system. Web pages that contain Java **applets** include the **applets** through the use of the proposed HTML tag <APP> (a third party extension to HTML proposed by Sun Microsystems) [10]. This HTML tag refers the Web browser to the location of the **applet** and its code. A Java enabled Web browser (which is platform-dependent software) reacts to an <APP> tag by downloading the Java class that is referenced in the tag, then downloading any additional classes that the referenced class requires. Once the classes are downloaded (a typically speedy process given the typically small Java classes), the browser executes the code through an internal Java interpreter (which is the software that simulates the Java Virtual Machine). Java is often referred to as an “interpreted and compiled” language. There is a small performance penalty resulting from the required code interpretation. This is somewhat of an unavoidable problem, but not one which should adversely affect the use of Java. As long as there are differing CPUS and operating systems, there will be native software which will take advantage of those differences. The only Web browser that currently supports Java is Netscape Navigator 2.0 [5].

Version 1.0 of the Java Development Kit (JDK) appeared in mid January 1995. Versions of the JDK are available for the Sun Spare Solaris, Microsoft Windows NT, and Microsoft Windows 95 operating systems [11]. The Macintosh (MacOS) JDK Beta 1 release appeared in mid February and the authors are not aware of any upcoming JDK for Microsoft Windows 3.11. JDK 1.0 includes: Java classes, source, compiler, interpreter, **appletviewer**, debugger, documentation generator, and Application Programming Interface (API) documents. The most complete archive of Java **applets** is **Gamelan**, at <http://www.gamelan.corn/>. Official information and links to various Java **applets** can be found at <http://www.javasoft.com/applets/index.html>. The only USENET newsgroup currently existing for Java is **comp.lang.java**. While the documentation distributed with the JDK is not enough to begin using Java, excellent tutorials are available at many sites (almost all of which may be found through **Gamelan**). Mailing lists on Java topics are listed at http://www.javasoft.com/Mail/external_lists.html.

3. ADVANTAGES AND DISADVANTAGES OF USING JAVA

The authors feel that Java has the following advantages for the development of **software** in an academic environment:

- **Tailored for the Internet** - The Java language is unique in its in-built handling of the network protocols required to send and receive data across networks such as the Internet. For the programmer, Java has made accessing a file stored remotely (on a Web or **FTP** server) as easy as accessing a file stored locally (on a



user's floppy or hard disk). Programs can easily be written to allow world-wide access to and interaction with various information resources.

- **Platform Independent** - The major strength of Java is that most software developers are no longer forced to choose between computer platforms, or expend great resources porting code from one platform to another. Further, code/class resources developed by others are no longer unavailable for use by programmers simply because the platform on which they were developed is different. The authors find platform independence **extremely** attractive. Academic communities are littered with a vast assortment of computers and operating systems. Personal preference and differing computing needs have led to campuses with operating systems which have included MS Windows, MacOS, **Solaris** and Motif X-Windows. Educators have scant resources enough that supporting more than one platform is impossible. Choosing one platform means excluding others and thereby limiting the usefulness of the resource to students and contributing faculty. Beyond the local campuses, platform independence greatly increases the usefulness of developed resources to the world-wide Internet community.
- **Interactive Content** - Java is making Web pages dynamic. User input can come through all of the user's familiar computing tools: keyboard, mouse, and in the future through computer's microphone.
- **Computation Aspects** - Java allows a Web environment in which numerical analysis can be performed and solutions found to **computationally** intensive engineering problems. Users will have an environment for **open**-ended interaction rather than the fixed set of options currently available through fill-out forms and CGI programs.
- **Java Applications Run Locally** - Java runs the Java **applets** locally, rather than requiring a constant and slow continual transfer of data from a remotely-located Web server. An **applet** is downloaded in its entirety before it is run; this does not mean that data cannot be passed between the **applet** and remote data servers as the **applet** is run, it simply means that the Web server on which the Java code runs is not the site of the program's responses. The old method of interaction, CGI programs, required that the Web server computer make every decision; with this arrangement, interactive Web content was both difficult to manage and slow.
- **Security** - Security is a significant concern whenever files can be sent and received from anywhere. Harm can come to a computer in many forms, including a computer virus, a Trojan horse, or a program which corrupts (intentionally or unintentionally) the computer's memory. The Java language and program **environment** has tried to address these concerns by restricting the ways in which **mis-aligned** programmer can access the host computer on which his/her code is executed.
- Some other advantages are discussed in [12].

Some disadvantages of using Java include the following:

- **Documentation** - The current state of Java documentation is less than ideal. At this stage, a person wanting to learn Java must engage **themselves** in a world-wide search for the necessary documentation. The documentation is available and basic tutorials are written and spread across the Web. The books published thus far are more introductory than comprehensive [13]; someone interested in learning the Java language would be better spending their money and their time on Web access rather than the available books. This is not uncommon for a new software technology where code often is ahead of documentation. To address this problem Sun Microsystems, Inc. has recently teamed up with Addison-Wesley Publishing Company to publish "The Java Series" [14] with titles having appearing in December 1995. The Web will continue to be the best source for the most up-to-date information.
- **Programming Knowledge** - Java is a programming language; programming experience and/or great patience and free time is required to turn ideas into Java code. Java is heavily based upon C++. Anyone familiar with C++, or any object-oriented programming language, will **find** the transition to Java programming fairly natural [15]. JavaScript [16] has been developed by Sun as a scripting language that allows many of the



benefits of Java without the programming knowledge or experience required for the Java language. JavaScript is perfect for such tasks as developing robust fall-out forms. The capabilities of JavaScript are quite limited. The novice programmer may find JavaScript to be a useful tool. The power of Java will become increasingly available to novice users as Java programming tools become more available. Visual tools, visual resource workshops and integrated development environments are available, or soon to be made available, from Symantec [17], Borland [18], and Sun [10].

- **Performance Concerns** - *Since* Java must be interpreted as it is executed, there are valid concerns regarding the code's execution speed. Various remedies for this have been proposed. The most reasonable and probable solution to this problem is pass the compiled code through a native optimizing program which would create a natively executable code from the platform independent classes. In this way, platform independence is maintained and performance is improved.
- . **Available Code** - *Since* Java is new, third-party classes/programs are lacking. Programmers working in Java must write much of their code for such routine tasks as plotting output; this is common for any new programming language.
- . Some other disadvantages of Java are addressed in [12].

4. USING JAVA TO DEVELOP INTERACTIVE INSTRUCTIONAL COURSEWARE

Educators have been using the Web this past academic year in courses to post course guidelines, homework, etc., and to develop courseware (instructional modules) [7, 8]. Instructional courseware that can be found on the WWW today is almost completely static and offers no significant advantages over textbooks. New advances in technology demand an evolution from this out-dated approach to instruction. The Java language has given the Web the power of continuous, interactive, real-time, visual and aural instruction.

When accessing a Web page, instead of clicking on a passive screen of images and text the user can be prompted for a response by offering an interactive menu while on-line instructions are delivered through an audio/animation sequence. The method of presenting courseware will pique students interests and understanding of the course material. The rich multi-media flavor of CD-ROMs can now be found in the academic/corporate networks and the Internet [19].

To both demonstrate and test Java's usefulness in an academic environment, the authors have developed a Java applet that allows a student to learn the steps involved in the stiffness matrix assembly process found in the finite element method (Section 5). A student using Netscape Navigator 2.0 could load the finite element method tool and interact with it through an input device such as a mouse.

5. A JAVA APPLET EXAMPLE FOR THE FINITE ELEMENT METHOD

A matrix assembly applet was developed to demonstrate how Java can be used to develop an interactive learning environment for the finite element method on the WWW. The matrix assembly applet can be found at location <http://amber.wpi.edu/~FEMur/Interactive-Learning-Tools/>. Java development kit 1.0 for Windows 95 [11] was used to develop the applet herein. A Java-enabled Web browser such as Netscape Navigator 2.0 [5] is required to run the applet on the Web. If your browser does not support Java, nothing will happen when you run the matrix assembly applet. The authors would like to emphasize that the matrix assembly applet can be used on any computer platform, i.e., MS Windows 95, MS Windows NT, Macintosh MacOS, Sun Spare Solaris, as long as you have a Java-enabled browser for your platform. A listing of assembly applet can be found at

<http://amber.wpi.edu/~FEMur/Conference-Papers/ASEE96/>. The processing power of a 486 66MHz computer with 8 Mb of RAM is sufficient to run the **applet**.

Consider the example problem in Figure 1 where steel columns are used to support the symmetric loads from the two floors of a building. A finite element model consists of two one-dimensional bar elements as shown in Figure 2. Buckling is neglected in this example. The student will use the Java-enabled Web browser to go through the interactive stiffness assembly procedure. Once the appropriate HTML document is loaded, the assembly **applet** is executed on the host computer and on the terminal screen the student is presented with the element stiffness matrices. The matrix assemble **applet** is shown in Figure 3. The force-displacement relationships of elements 1 and 2 are shown in Figure 3 in terms of global node numbers. The element stiffness matrix is of order 2×2 since it relates two nodal forces to two nodal displacements, i.e., $\underline{f}_e = \underline{K}_e \underline{u}_e$. The student then defines the size of the global force-displacement relationship for the problem. **Since** the mesh contains 3 nodes with one degree of freedom per node, the force-displacement equation relates 3 nodal forces to 3 nodal displacements through a 3×3 global stiffness matrix, i.e., $\underline{f} = \underline{K} \underline{u}$. Once the equation size is defined by the student, the Java **applet** will generate the empty global force-displacement relationship as shown in Figure 4.

The stiffness matrix \underline{K} of the structure under consideration can be found by using the direct assembly method [20]. It states that the contribution of all elements are simply added into the global **stiffness** matrix according to the degrees of **freedom** (node numbers in this case) associated with the elements. This process leads the formation of the global stiffness matrix for the entire structure. The assembly process using the **applet** is carried out by dragging an element stiffness term and dropping it into the appropriate position in the global stiffness matrix. The drag and drop feature of the learning environment is shown in Figure 5.

The matrix assembly **applet** has been developed with the following educational features in mind:

- **Interactive Learning Capabilities** - When an element stiffness term is assembled into the global stiffness matrix the student is informed whether or not the term was placed correctly. If an incorrect assembly is done then student has the option of trying again or allowing the software to show the correct location. A comment is also provided by the software so that more insight can be obtained by the student. The user can also request that the assembly process be completely done by the software on an element stiffness term basis or element stiffness matrix basis.
- **Color Coding of Element Stiffness Matrices** - Each element stiffness matrix has been assigned a unique color so that the student many see its effect and placement in the global stiffness matrix. When stiffness terms from different elements are added together (combined) in the global **stiffness** matrix the color of the matrix position is represented by a combination of all element colors. The restart button can be used to restart the assembly process.
- **Zero Stiffness Terms** - The 'zero stiffness term' button can be used to add a zero **stiffness** term on a individual basis into the global stiffness matrix. Feedback is provided to inform the student if the position is correct or incorrect. The 'zero out all stiffness terms' button will zero out all unoccupied and non-zero positions in the global stiffness matrix.
- **Developer's Feedback** - The feedback button provides a electronic mechanism for the student to notify developers regarding any technical and content problems that arise during the execution of the **applet**.
- **On-line Help** - On-line help has been included if any additional information or **clarification** is needed regarding the features described above.

6. FUTURE DEVELOPMENT OF AN INTERACTIVE LEARNING TOOL FOR THE FINITE ELEMENT METHOD

The author's are currently developing a Web site for the finite element method called the Finite Element Method universal resource (FEMur) and its Internet address is <http://amber.wpi.edu/~FEMur/>. The FEMur homepage is shown in Figure 6. FEMur will contain the following two components: 1. Resources for FEM on WWW.; 2. Learn the Finite Element Method.

'Resources for FEM on WWW' is a collection of organized links to FEM on the World-Wide Web. Some resources include commercial FEM codes, textbooks, other FEM homepages, consulting services, newsgroups, university, government and personal homepages, and professional societies. *'Learn the Finite Element Method'* is a modular resource that will include theoretical aspects and design/analysis applications of an element in an electronic textbook form. The author's are currently developing a prototype FEM interactive learning tool for the one-dimensional bar element using Java that will be integrated into the *'Learn the Finite Element Method'* component of FEMur. The interactive learning tool for finite element method is called FEMur-CAL (Finite Element Method universal resource Computer-Assisted Learning). The matrix assembly applet in Section 5 demonstrates on a small scale how Java can be used to develop an interactive learning environment for the finite element method that has a friendly graphical user interface (GUI) and on-line help. At the ASEE conference the author's will demonstrate an alpha version of the interactive FEM learning tool FEMur-CAL. Future updates of the FEM interactive learning tool can be found at URL address <http://amber.wpi.edu/~FEMur/Conference-Papers/ASEE96/>.

The student will begin an on-line interactive session in FEMur-CAL by starting with the problem definition and the mesh topology. Then the student will proceed to solve the problem in a step-by-step manner according to FEM. Fill-out forms will be used to implement the interactive interface within the WWW browser. The student will also go through the process of working with and solving the finite element matrix equations on-line. Results will be displayed in graphical and tabular forms.

The authors would like to emphasize that the objective of FEMur-CAL is to be an learning tool and it is not intended to be used for commercial purposes in analyzing and designing industrial size problems. Furthermore, FEMur-CAL is neither a stand-alone teaching product nor a substitute for the conventional classroom experience. FEMur-CAL is not only intended to supplement classroom work, but will also be available for examination review process and make-up classes.

7. CONCLUSION

The Java programming methodology is very new and it is currently in Version 1.0, therefore, the current support for its applet programming and the applet viewing in the WWW browsers still has bugs and inconsistencies. However, it supports a new paradigm in computer programming by allowing to write programs that are computer platform independent. It also provides explicit support for network oriented computing, which improves the accessibility of the information by the user and in the recent years gaining momentum through the development of global computer networks such as Internet.

At the time of writing this paper Version 1.0 of Java Development Kit (JDK) has been released by Sun Microsystems Inc. in mid January 1996, and many example applets/stand-alone programs have appeared on the Web [21]. The future of development effort on the Web utilizing Java appears to be very promising. Since its introduction in summer of 1995 major software development companies such as Oracle, Microsoft [22], and

IBM [23] have licensed the technology to integrate its programming methodology in their WWW based software products.

This paper provides an overview of Java programming language and its use in the development of interactive content for the WWW. A demonstration **applet** using Java language has been developed for the stiffness assembly procedure. It will be used as an interactive learning tool for the finite element method. The **applet** program can be executed at any location through a computer connected to Internet and running a **Java**-enabled Web browser by accessing the URL <http://amber.wpi.edu/~FEMur/>. The Web site also provides information regarding Finite Element Method and a list of general resources. Up-to-date information regarding this paper can be found at <http://amber.wpi.edu/~FEMur/Conference-Papers/ASEE96/>.

ACKNOWLEDGEMENT

The authors would like to thank Peter Kwok (kwok@wpi.edu), a junior undergraduate Mechanical Engineering student at WPI and a member of the FEMur development team, for drawing the figures for the example problem in Section 5.

REFERENCES

1. Hughes, K., "Entering the World-Wide Web: A Guide to Cyberspace;" Enterprise Integration Technologies, May 1994, <http://www.eit.com/web/www.guide/>.
2. Hughes, K., "From Webspace to Cyberspace," Enterprise Integration Technologies, Version 1.1, July 1995, <http://www.eit.com/~kevinh/cspace/>.
3. Neilson, J., Hypertext and Hypermedia, Academic Press Inc., 1994.
4. Balasubramanian, V., "State of the Art Review on Hypermedia Issues and Applications," Graduate School of Management, Rutgers University, Newark, New Jersey, March 1994, http://www.isg.sfu.ca/~duchier/misc/hypertext_review/index.html.
5. Homepage for Netscape Navigator Browser, <http://home.netscape.com/>.
6. Download for NCSA Mosaic, <http://www.ncsa.uiuc.edu/SDG/Software/SDGSoftDir.html>.
7. Shawki, T. G., "TAM221 Mechanics of Materials," Department of Theoretical and Applied Mechanics, University of Illinois at Urbana-Champaign, Champaign, Illinois, May, 1995, <http://e2.tam.uiuc.edu/TAM221/index/>.
8. Kayser, J., "Statics Tutorial," Department of Civil Engineering, Lafayette College, Easton, Pennsylvania, 1995, <http://www.lafayette.edu/kayserj/statics/cover.htm>.
9. Martin, T., "The Development of Interactive World-Wide Web Courseware for Students of Engineering and Technology at Deakin University," Faculty of Science and Technology, School of Engineering and Technology, Deakin University, 1995, <http://www.scu.edu.au/ausweb95/papers/education1/martin/>.
10. The Java Language, Sun Microsystems, Inc., Mtn. View, California, 1995, <http://www.javasoft.com/about.html>.
11. Downloading the Java Developers Kit, Sun Microsystems, Inc., Mtn. View, California, 1995, <http://www.javasoft.com/download.html>.
12. Gosling, J. and Henry McGilton, H., "The Java(tm) Language Environment: A White Paper," Sun Microsystems, Inc., Mtn. View, California, 1995, <http://www.javasoft.com/whitePaper/java-whitepaper-1.html>.
13. BeaconRay's Great Books, THE JAVA SHELF, <http://jollyroger.com/java.html>.
14. The Java™ Series, Addison-Wesley Publishing Company, <http://www.aw.com/cp/javaseries.html>.

15. Harold, E. R., "Brewing Java: A Tutorial," January, 1996, <http://sunsite.unc.edu/javafaq/javatutorial.html>.
16. JavaScript, http://home.netscape.com/comprod/products/navigator/version_2.0/script/.
17. Symantec Cafe, <http://cafe.symantec.com/cafe/index.html>.
18. Borland Debugger (TM) for Java (TM), <http://www.borland.com/Product/java/debugger/bugform.html>.
19. Winslow, M. and Hamner, D., "Java and the WWW," North Carolina System Administrators Lecture Series, October, 1995, <http://www.cs.unc.edu/~winslow/talk/outline.html>.
20. Grandin, Jr., H. T., Fundamentals of Finite Element Method, Waveland Press, Inc., Prospect Height, Illinois, 1991.
21. "Gamelan: The Directory and Registry of Java Resources," EarthWeb, <http://www.gamelan.com/>.
22. Green, H., "Microsoft to License Sun Java Program," New York Times Syndicate, December 7, 1995, http://nytsyn.com/live/News3/341_120795_1_34_125_6450.html.
23. IBM Corporation, "IBM licenses Java technology from Sun Microsystems for use in Internet products," Press release, December 6, 1995, <http://www.ibm.com/News/javapr.html>.

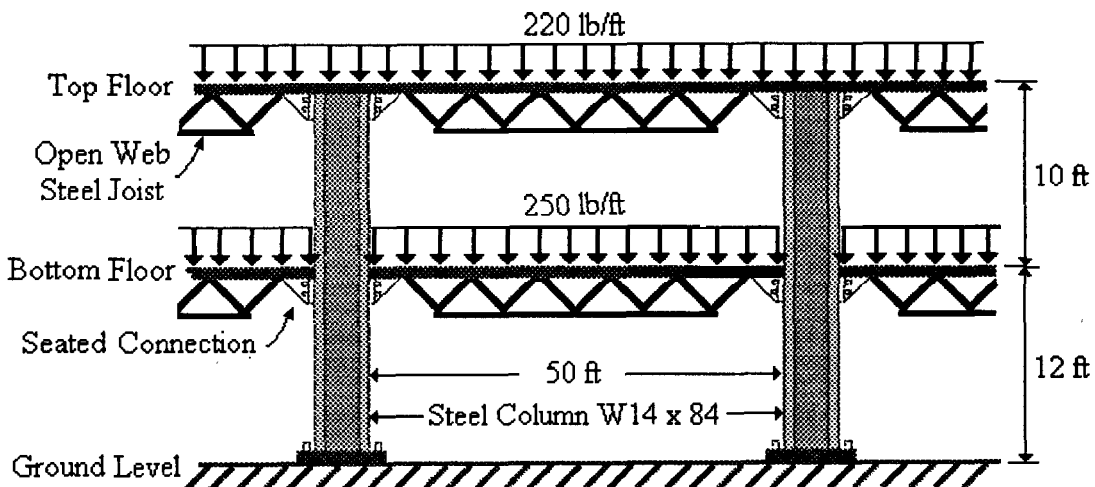


Figure 1. Problem definition.

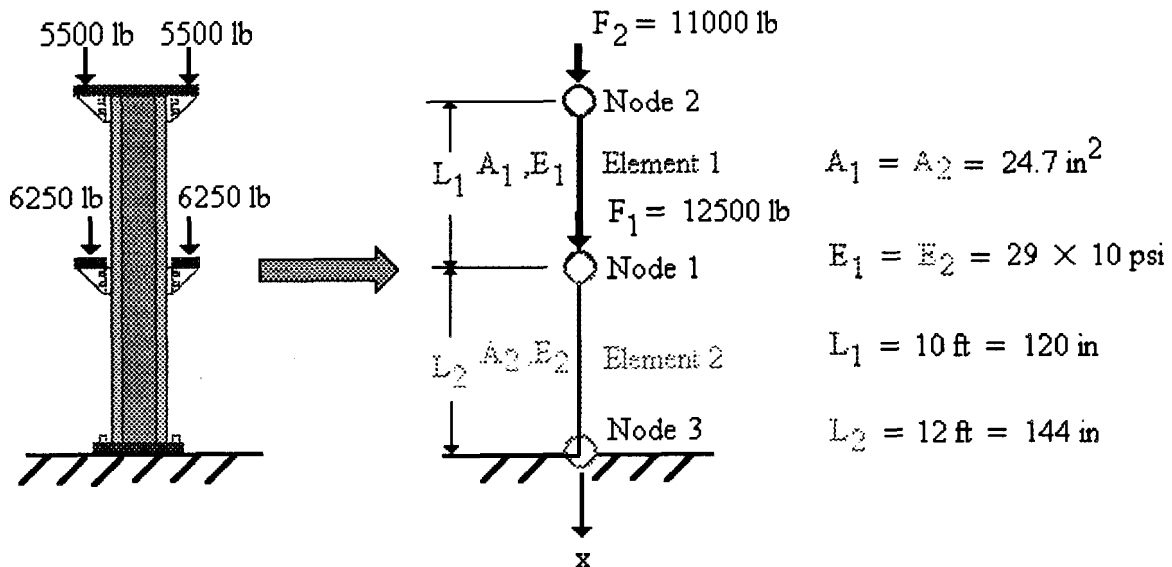


Figure 2. Finite element mesh.

FEMur-CAL presents Assemble-It!

Element 1:

$$\begin{Bmatrix} f_2 \\ f_1 \end{Bmatrix} = \begin{bmatrix} & 2 & & & 1 \\ 6621666.67 & & -6621666.67 & & \\ & & & & \\ -6621666.67 & & & & 6621666.67 \\ & & & & \end{bmatrix} \begin{matrix} 2 \\ 1 \end{matrix} \begin{Bmatrix} u_2 \\ u_1 \end{Bmatrix}$$

Element 2:

$$\begin{Bmatrix} f_1 \\ f_3 \end{Bmatrix} = \begin{bmatrix} & & & & 1 \\ & & & & \\ 5518055.56 & & -5518055.56 & & \\ & & & & \\ -5518055.56 & & & & 5518055.56 \\ & & & & \end{bmatrix} \begin{matrix} 1 \\ 3 \end{matrix} \begin{Bmatrix} u_1 \\ u_3 \end{Bmatrix}$$

The force-displacement relationship of the entire structure relates nodal forces to nodal displacements through a stiffness matrix of order x

Symbolic Form:

$$\underline{F} = \underline{K} \underline{u}$$

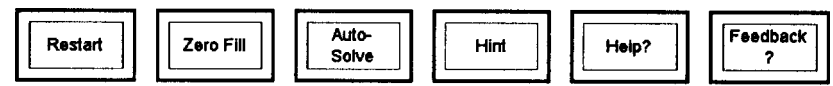


Figure 3. FEMur-CAL matrix assembly applet.

FEMur-CAL presents Assemble-It!

Element 1:

$$\begin{Bmatrix} f_2 \\ f_1 \end{Bmatrix} = \begin{bmatrix} & 2 & & & 1 \\ 6621666.67 & & -6621666.67 & & \\ & & & & \\ -6621666.67 & & & & 6621666.67 \\ & & & & \end{bmatrix} \begin{matrix} 2 \\ 1 \end{matrix} \begin{Bmatrix} u_2 \\ u_1 \end{Bmatrix}$$

Element 2:

$$\begin{Bmatrix} f_1 \\ f_3 \end{Bmatrix} = \begin{bmatrix} & & & & 1 \\ & & & & \\ 5518055.56 & & -5518055.56 & & \\ & & & & \\ -5518055.56 & & & & 5518055.56 \\ & & & & \end{bmatrix} \begin{matrix} 1 \\ 3 \end{matrix} \begin{Bmatrix} u_1 \\ u_3 \end{Bmatrix}$$

The force-displacement relationship of the entire structure relates nodal forces to nodal displacements through a stiffness matrix of order x

Symbolic Form:

$$\underline{F} = \underline{K} \underline{u}$$

Expanded Form:

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} = \begin{bmatrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix} \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

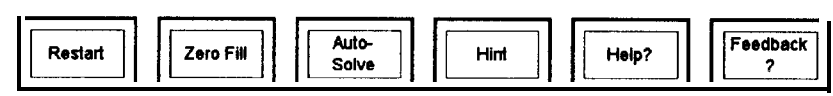


Figure 4. Empty global force-displacement relationship.

FEMur-CAL
presents
Assemble-It!

Element 1:
$$\begin{Bmatrix} f_2 \\ f_1 \end{Bmatrix} = \begin{bmatrix} 6621666.67 & 0 \\ 0 & 6621666.67 \end{bmatrix} \begin{Bmatrix} u_2 \\ u_1 \end{Bmatrix}$$

Element 2:
$$\begin{Bmatrix} f_1 \\ f_3 \end{Bmatrix} = \begin{bmatrix} 5518055.56 & 0 \\ 0 & -5518055.56 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_3 \end{Bmatrix}$$

The force-displacement relationship of the entire structure relates nodal forces to nodal displacements through a stiffness matrix of order x

Symbolic Form:

$$F = K u$$

Expanded Form:

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \end{Bmatrix} = \begin{bmatrix} 6621666.67 & 0 & 0 \\ 0 & -6621666.67 & 0 \\ 0 & 0 & 5518055.56 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix}$$

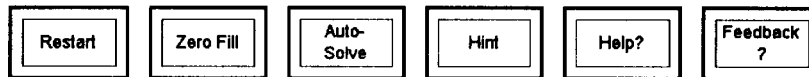


Figure 5. Drag and drop feature.

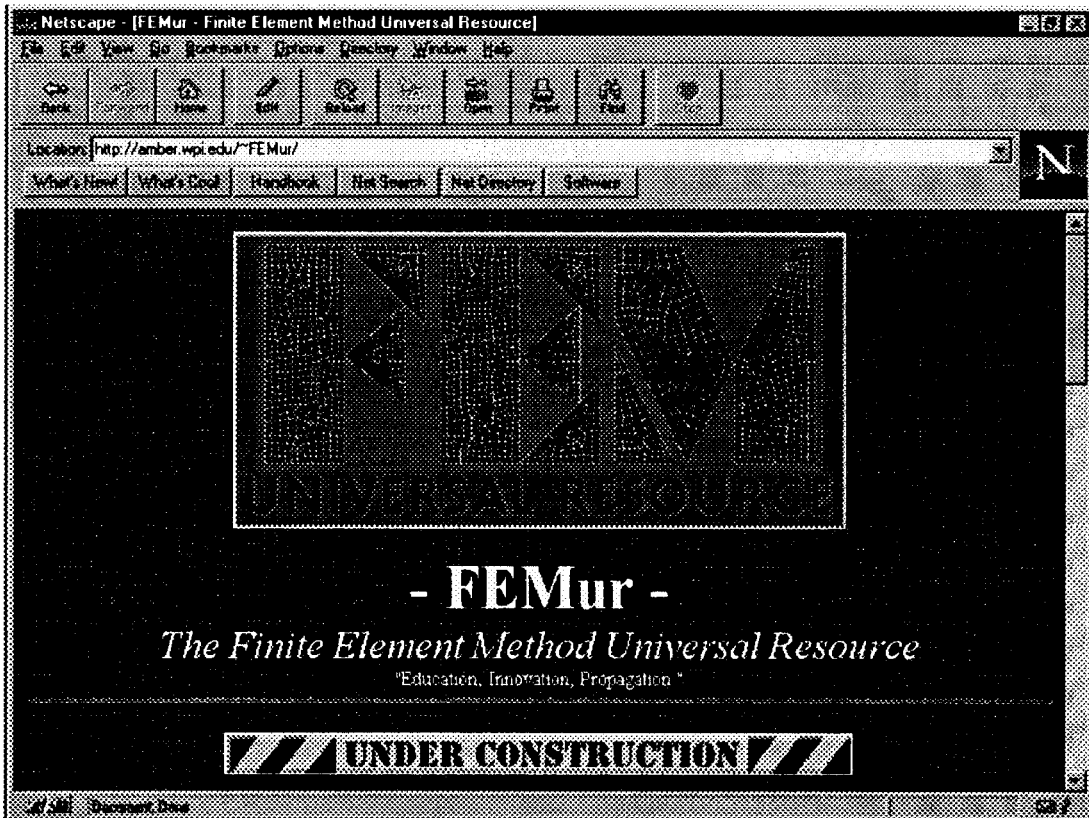


Figure 6. FEMur Homepage.

BENJAMIN “QUINCY” CABELL V

Benjamin “Quincy” Cabell V received his B.S. from Worcester Polytechnic Institute in Mechanical Engineering. He is currently pursuing graduate school at the same institution, hoping to further develop his skills in mechanical design, writing software for instructional purposes, and developing new software for use by mechanical engineers.

JOSEPH J. RENCIS

Joseph J. Rencis is an Associate Professor in the Mechanical Engineering Department at Worcester Polytechnic Institute. His research focuses on the development of boundary and finite element methods for analyzing solid, heat transfer and fluid mechanics problems. He received his B.S. from the Milwaukee School of Engineering in 1980, a M.S. from Northwestern University in 1982 and a Ph.D. from Case Western Reserve University in 1985.

JAVED ALAM

Javed Alam is a professor of Civil and Environmental Engineering at Youngstown State University. He obtained his B.S. in Civil Engineering from Indian Institute of Technology at Kanpur, India and received his M.S. from Asian Institute of Technology at Bangkok, Thailand. He pursued further studies at Case Western Reserve University in Cleveland, Ohio to obtain a Ph.D. degree. His research interests are in the area of Structural Mechanics and Computer Applications in teaching.