

3-D Finite Element Analysis on the Internet using Java and VRML

Karthik Ranga and Kurt Gramoll
The University of Oklahoma
School of Aerospace and Mechanical Engineering

Abstract

Not only has the Internet gained a prominent role in the industry by revolutionizing commerce, information access, supply-chain management, entertainment and publicity etc but it is also quickly impacting engineering design and technical education. This paper describes the research conducted by the Engineering Media Lab at the University of Oklahoma in applying Java and VRML (Virtual Reality Modeling Language)¹ for performing engineering design and analysis on the Internet.

To demonstrate the potential of using VRML and Java for design and analysis over the Internet an L-shaped beam subjected to a point load along any or a combination of three directions i.e. X, Y and Z is chosen. The user can vary the length and height of the beam. For the sake of application simplicity the cross-section of the beam is constant. The user can choose the material properties of the beam and the loads to which the beam is subjected. After the user submits the values, the Java program generates the new L-beam, meshes the beam automatically and computes the 3-D stresses in the beam using the finite element method. The results are color coded and displayed on the screen. Thus the user can perform the design and analysis from any computer with an Internet connection in a virtual design environment with 3-D visualization.

1. Introduction

The Internet has become one of the fastest growing technologies to impact human society. The number of users embracing the Internet is increasing exponentially. The Internet is now being widely used for a variety of common tasks such as sending e-mails, listening to music, obtaining news and information and playing games. Over the last several years the Internet has also revolutionized the business sector by defining a new way of conducting business and trade known as e-commerce. The growth of e-commerce into a multi-billion dollar entity has evoked a great interest in the Internet from almost every enterprise. The Internet is also being widely used for conducting on-line classes in many universities².

The ubiquitous, inexpensive and platform independence nature of the Internet are some of the reasons for its increasing popularity. People are trying to use the Internet for newer and better tasks. One of the areas where the Internet can be used extensively is in the field of mechanical engineering design. The strong competition among the designers and manufacturers of various products require better products be designed and marketed quicker than ever before. This puts pressure on companies to employ various new technologies in integrating both the product development and production processes to enable the product to reach the customers in a short time and thus compete in the global economy. It is of benefit to the product development organizations to explore and come to understand the function, capabilities and potential usefulness of the Internet³. The ability of the Internet to share information anywhere anytime and with anyone makes it ideally suited for design and analysis where the designers maybe located in different physical locations. Thus the companies can leverage the ubiquitous strength of the Internet for conducting design and analysis of their products⁴.

The objective of the research described in this paper is to develop an Internet based design and finite-element analysis environment that includes 3-D visualization for use with engineering education. This research utilizes VRML for visualizing different designs and analyses of a beam on the Internet that can be demonstrated in a class or virtual discussion on the Internet. A virtual design environment is created in which three-dimensional models can be visualized over the Internet using a web browser and a VRML plug-in. The user who has an Internet connection to his or her computer can modify the various attributes of the design model. The model used to demonstrate web-based design and analysis in this research is an L-shaped beam fixed at one end. The stress analysis is performed using the finite element analysis method.

2. VRML

Since the primary objective of this research is to demonstrate design and analysis on the Internet using a virtual design environment it was essential to use a 3-D file format, which can be viewed on the Internet. VRML is the most popular standard for representing 3-D objects on the Internet. VRML is an acronym for the Virtual Reality Modeling Language⁵. Technically speaking, VRML provides a bare minimum of geometric modeling features and contains numerous features. At its core, VRML is simply a 3D-interchange format. It defines most of the commonly used semantics found in today's 3D applications such as hierarchical transformations, light sources, viewpoints, geometry, animation, fog, material properties and texture mapping^{6,7}. One of the primary goals in designing VRML was to ensure that it at least succeeded as an effective 3D file interchange format. VRML can be considered as a 3D analog to HTML. This implies that VRML serves as a simple multiplatform language for publishing 3D web pages. This is motivated by the fact that some information is best-experienced three dimensionally, such as games, engineering and scientific visualizations, educational experiences and architecture. Typically these types of projects require intensive interaction, animation and user participation and exploration beyond what is capable with HTML format. VRML provides the technology that integrates 3D, 2D, text and multimedia into a coherent model. When these media types are combined with scripting languages and Internet capabilities, an entirely new genre of interactive applications is possible³.

VRML is designed to fit into the existing infrastructure of the Internet. It uses existing standards wherever possible, even if those standards have some shortcomings when used with VRML. Using existing standards instead of inventing new, incompatible standards makes it much easier for the Web developer, who can use existing tools to help create VRML content. VRML files may contain references to files in many other standard formats. JPEG, PNG, GIF and MPEG files may be used as texture maps on objects. WAV and MIDI files may be used to specify sound that is emitted from in the world. Files containing Java or JavaScript code maybe referenced and used to implement programmed behavior for the objects in the VRML worlds.

3. JAVA

The programming language used to control the VRML beam and perform the finite element analysis requires seamless integration with the Internet and the web browser. These requirements are satisfied by the Java programming language developed by Sun Microsystems^{8,9}. Java has become very popular because of the many features designed to make it operate on the Internet. Developers can write full-fledged applications in Java, whose architecture is much like C and C++. Small applications called applets can also be written in Java and are designed to be downloaded over the Internet and used in conjunction with Web pages. Such Java applets are designed to operate in a restricted area to prevent them from causing any permanent damage to the host computer. Otherwise, Java applets can do almost anything a standard desktop application can do, including operating interactively with the user, and interacting with resources on the originating server, such as databases. Java applications are platform-neutral¹⁰. They can run on any platform that has a Java Run Time Engine or just in time Java compiler. Java holds tremendous potential for scores of diverse applications, but many are still in the development stage. As the Java language matures its use will become more pervasive, especially in the large enterprises. For the product development organization, there are many uses for Java applications.

4. Finite Element Method

The finite element method is a numerical method for solving problems of engineering and mathematical physics^{11, 12, 13}. For over 30 years it has been a powerful tool for the numerical solution of a wide range of engineering problems from deformation and stress analysis of automotive, aircraft, building and bridge structures to field analysis of heat flux, fluid flow, magnetic flux and other flow problems⁵.

For problems involving complicated geometries, loadings and material properties, it is generally not possible to obtain analytical mathematical solutions. Analytical solutions are those given by a mathematical expression that yields the values of the desired unknown quantities at any location in the body and are thus valid for an infinite number of locations in the body. These analytical solutions generally require the solution of ordinary or partial differential equations, which, because of the complicated geometries, loadings and material properties, are not easily obtainable. Hence, we need to rely on numerical methods such as the finite element methods for an acceptable solution. The finite element formulation of the problem results in a system of simultaneous algebraic equations for solution, rather than differential equations. This method

yields the approximate values of the unknowns at discrete numbers of points in the continuum. Hence this method of dividing a body into an equivalent system of smaller bodies interconnected at their respective boundaries is called discretization. In the finite element method, instead of solving the problem for the entire body in one operation, we formulate the equation for each finite element and combine them to obtain the solution for the whole body.

5. Finite Element Analysis using VRML and Java

Having described the various background technologies used in this research it is now possible to explain how these technologies were combined to achieve the end result of performing the finite element analysis of a simple three-dimensional object on the Internet. The following sub-sections describe the method of integrating Java and VRML, the choice of the tetrahedral finite element and the design and analysis of the L-shaped beam.

5.1 Integrating Java and VRML

VRML files describe 3D objects and worlds using a hierarchical scene graph. Entities in the scene graph are called nodes. Nodes store their data in fields, and VRML defines many types of fields that can store everything from a single value to an array of values. The VRML nodes can contain other nodes i.e., the nodes can have child nodes. To allow one node to communicate with the other nodes in the VRML worlds there is an event or message passing mechanism. Each node type defines the names and types of events that instances of that type may generate or receive, and Route statements define event paths between event generators and receivers. Script nodes can be inserted between event generators and event receivers. Scripts allow the world creators to describe arbitrary behaviors, defined in either Java or JavaScript languages supported by the VRML external-authoring interface.

If Java is to be allowed to interact with VRML, it needs a mechanism to access fields, manipulate their values and return the results to the VRML world. VRML 2.0 does that by extending its execution model into the Java world by using a special node that bridges the two worlds. That bridge is called the Script node^{14, 15}. The Script node is similar to any other node in some respects but very powerful in other respects. The similarity is due to the fact that we can place it anywhere in the scene hierarchies and route events to and from the Script node like any other node. However, it differs in one key aspect: the fields of a Script node are user-extensible, and events arriving at those fields are automatically routed to the program associated with the Script node. The URL field in the Script node provides the link between the node and the program that will implement some behavior on behalf of this Script node. It enables the scene author to bind code to the Script node. The field is named URL because it is intended to hold a Uniform Resource Locator and signifies that this field usually contains a string that points to a file containing the program that will be used by this Script node. Although VRML 2.0 enables any programming language to implement its behaviors, the majority of serious VRML developers use Java – because of its power and tight integration with the Internet. The Script node also contains user-defined fields that provide the link between the state in the VRML scene and state in the Java program. There can be any number of these fields, and they can be any legal VRML 2.0 types. They can also be one of the following: normal fields, eventIns and eventOuts.

They cannot be exposed fields. Any events arriving at an eventIn field automatically cause the browser to pass the event to the program referred to the URL field of the Script node. The method used by the browser to do this depends on the browser implementation and the language being called. This mechanism is specified for Java and is called event dispatching. It describes the way the event is dispatched to the piece of code that will handle the event. Any browser that supports Java must implement this event dispatching mechanism so that code for one browser will work for the other browsers too. After an event has been dispatched to the program, the program will handle that event. Eventually, the program will return some result from its computation. Again, the specification defines a way to enable the program to send information back to the Script node. Such information is written back to the fields in the Script node. This updates the value of the field in the Script node. Writing an eventIn or an eventOut has further effects. Writing to an eventIn in a Script node generates an event that is sent to the Java program. Writing to an eventOut field generates an event that will be sent to any nodes that are interconnected, via routes, to the eventOut field of the Script node.

Thus all that is happening in the model is that the Script node is acting as a messenger for the Script itself. Any events sent to the Script node are packaged and delivered to the Script. Any results written by the Java program update the Script node fields. Subsequently, the Script node will, if required, generate new events and automatically send them on. This means that the Script node can be placed anywhere in the event cascade. Whenever you need to do a little extra processing as a result of an event, route the event to a Script node, handle it in the script, and send the results back to the script node. If we want the results to also affect the other entities in the scene, send the result back to an eventOut field and have the Script node ROUTE the event to the next node in the chain.

5.2 Finite element analysis of an L-Beam

The model chosen for performing finite-element analysis is a simple L-shaped beam fixed at one end. The cross-sectional area of the beam is kept constant but the user can change the length and height of the beam. The user can also specify the point load at the top end of the beam in the X, Y and Z directions and the modulus of elasticity of the beam material. The window used to input the various design parameters is constructed using the Abstract Windowing Toolkit of Java.

This program uses the tetrahedral solid element for the finite element analysis. After the user submits the various input parameters for the beam, the program automatically generates the finite element mesh for the beam model by using a master cube element consisting of six tetrahedral solid elements. Since it is quite difficult to visualize a tetrahedral element the mesh generation algorithm uses this approach. It should be noted that the user does not have control over either the type of element or the mesh generation in order to make the 3-D analysis environment simpler to use. Upon submitting the values using the “FE Analysis” button, the Java program performs the computations on the local machine and updates the VRML model in the browser. The stresses are color coded and displayed on the VRML model for visual clarity. The average maximum and minimum stress values are also displayed on the VRML model. The back-end finite element analysis program written in Java is designed to compute the X, Y or Z stress values for the L-beam. The user selects the type of stress computations for the L-beam model by

selecting the appropriate radio button. The program has allows the user to either make the analysis model transparent or opaque by clicking on the appropriate radio button. When the user exits the browser window the last model does not get saved i.e., the user starts from the same model every time he opens the browser. Figure 1 shows the snapshot of the VRML design environment for a beam with the transparency turned off.

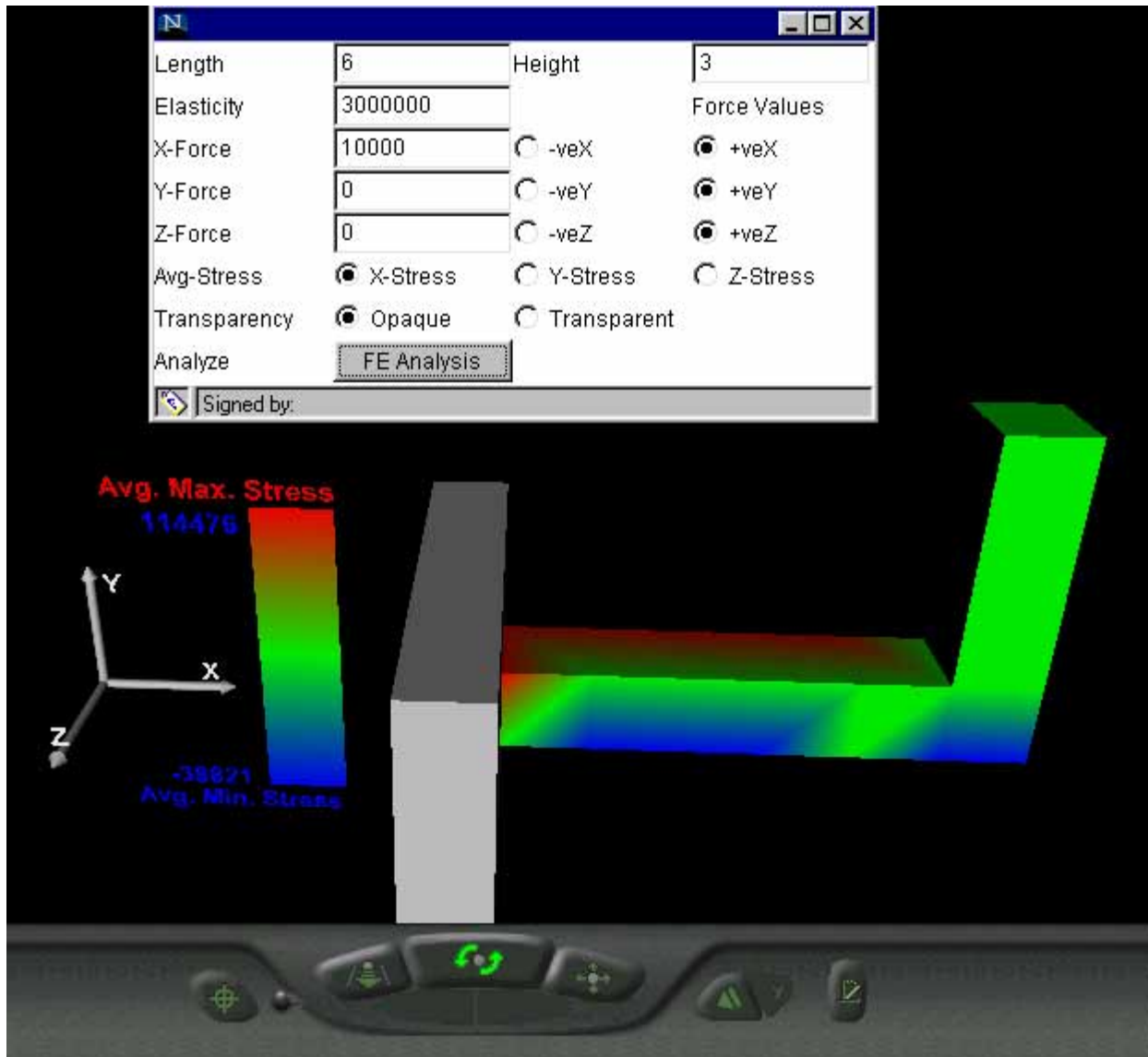


Fig. 1 L-beam with transparency turned off

In order to better visualize internal features of objects in 3-D VRML has transparency features built into it. In the L-beam design environment the user can make the beam transparent and thereby visualize the finite element mesh consisting of tetrahedral solid elements. Figure 2 depicts the snapshot of a beam with the transparency function turned on.

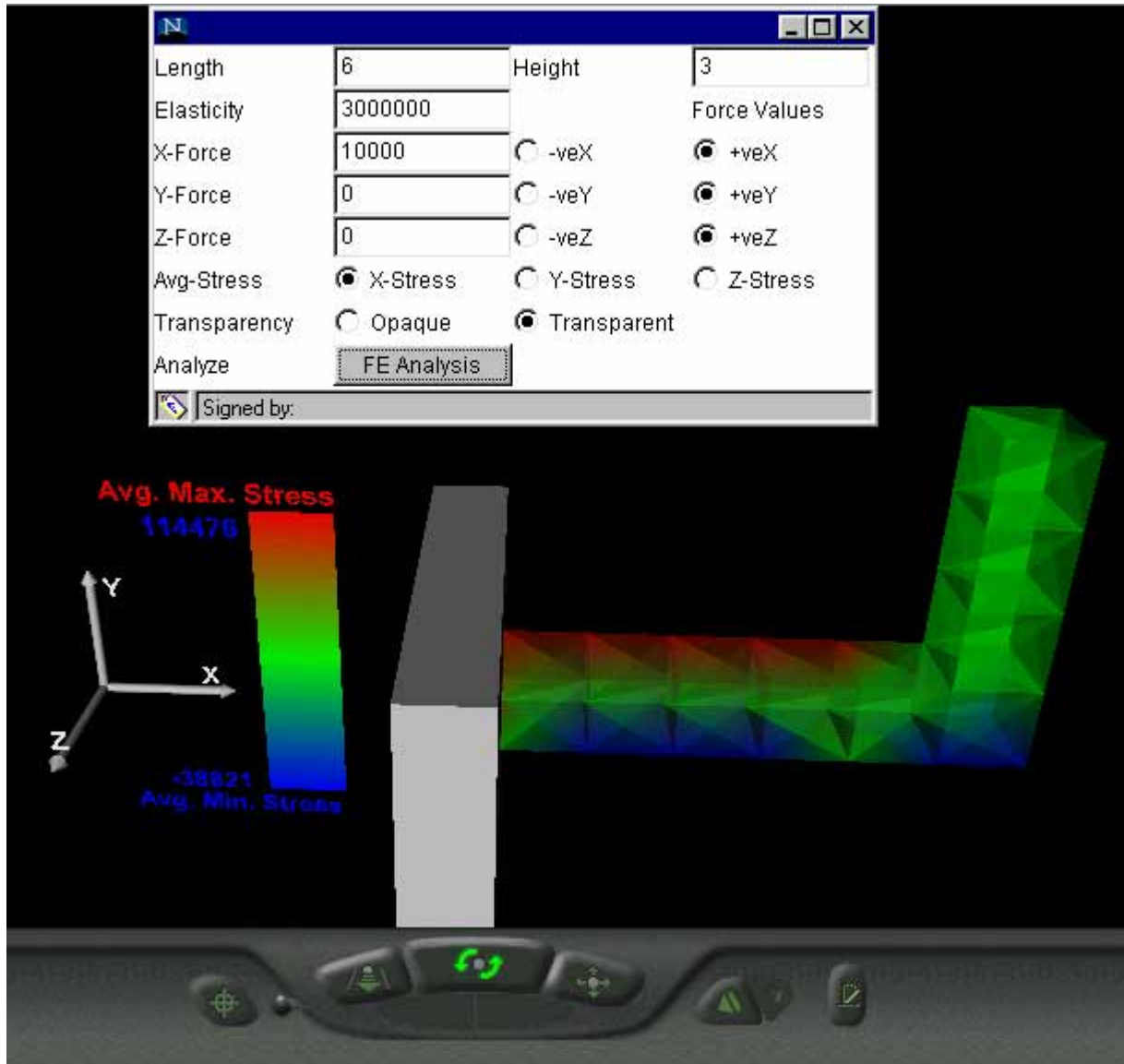


Fig. 2 L-beam with transparency turned on

6. Summary and conclusions

This paper demonstrates the potential for using the Internet for performing design and analysis particularly for imparting engineering education. The VRML technology used in this research is an open standard and inexpensive. However there are a number of limitations to the use of VRML. The main drawback of VRML is that it is a polygon based surface representation, and does not allow solid data such as mass, density etc required by modern 3-D CAD systems to be represented. Also it cannot model very complicated geometrical entities accurately. However, the future releases of VRML may support splines and curves. The Java programming language ties

in very well with the VRML standard and after learning the language the interested researcher can think up of many good ways to put the two technologies of VRML and Java for use over the Internet for engineering education.

References

1. Marrin, C., Campbell, B., Teach yourself VRML in 21 Days, Sams Publishing, 1997.
2. Gramoll, K., Teaching statics online with only electronic media on laptop computers, 1999 ASEE Conference and Exposition, Charlotte, NC.
3. Anthony Mills, Collaborative Engineering and the Internet, Society of Manufacturing Engineers, 1998.
4. Karthik, M., Concurrent-Interactive Design and Analysis using the Internet and VRML, Master of Science Thesis, The University of Oklahoma, Norman, 1998.
5. Ames, L., Nadeau, D.R., Moreland, J.L., VRML 2.0 Sourcebook, Second Edition, John Wiley, New York, 1997.
6. Hartman, J., Wernecke, J., The VRML 2.0 Handbook, Addison Wesley Publishing Company, 1996.
7. Carey, R., Bell, G., The Annotated VRML 2.0 Reference Manual, Addison Wesley Publishing House, 1997.
8. Naughton, P., The Java Handbook, Tata McGraw-Hill Publishing Company, 1997.
9. Lemay, L., Cadehead, R., Sams Teach Yourself Java 2 Platform in 21 Days, Sams Publishing, 1999.
10. Peter van der Linden, Just Java, The SunSoft Press Java Series, 1996.
11. Logan, L.D., A First Course in the Finite Element Method, First Edition, PWS-KENT, Massachusetts, 1992.
12. Chandrupatla, R., Belegundu, A., Introduction to Finite Elements in Engineering, Second Edition, Prentice Hall, 1997.
13. Segerlind, L., Applied Finite Element Analysis, John Wiley and Sons, 1976.
14. Lea, R., Matsuda, K., Miyashita, K., Java for 3D and VRML Worlds, New Riders Publishing, 1996.
15. McCarthy, M., Descartes, A., Reality Architecture – Building 3D Worlds with Java and VRML, Prentice Hall Europe, 1998.

Bibliographic

KARTHIK RANGA

Karthik Ranga is currently a Graduate Student in the Mechanical Engineering Department at the University of Oklahoma, Norman Campus. He received his B.S in Mechanical Engineering from the Karnatak University, India. Prior to returning to graduate school to obtain an advanced degree, he worked as a Design Engineer in the R and D department of Wipro Infotech Group, India.

KURT GRAMOLL

Kurt Gramoll is the Hughes Centennial Professor of Engineering and Director of the Engineering Media Lab at the University of Oklahoma. He has developed and published CDs and web-based sites for engineering education, K-12 instruction, and training in industry. He has started two multimedia companies for the development and distribution of technical electronic media. Dr. Gramoll received his B.S. degree in Civil engineering and M.S. degree in Mechanical Engineering, both from the University of Utah. He received his Ph.D. in Engineering Science and Mechanics from Virginia Tech. Previously, he has taught at the Univ. of Memphis and Georgia Tech.