Computing and Programming with LabVIEW

Shahnam Navaee
Georgia Southern University

Abstract

In this paper an alternate approach in teaching various computing and programming principles to engineering students utilizing the LabVIEW software is discussed. LabVIEW, acronym for "Laboratory Virtual Instrument Engineering Workbench", is a powerful and robust graphical programming language developed by National Instruments, Inc. for use in various engineering and scientific related fields. At the present time, this software tool is mainly utilized in the industrial, governmental, and educational venues to aid the investigators in conducting laboratory experimentations, and in collection and analysis of recorded data. The submitted paper specifically focuses on the programming capabilities of LabVIEW rather than the utility of this tool in data acquisition and laboratory measurements. All essential and important tools and constructs needed for developing solutions to classical engineering problems are included in LabVIEW. These include capabilities for setting up various conditional structures, loops and Case structures; special handling of array and matrix operations; solving systems of linear algebraic equations; solving differential equations; differentiating and integrating functions; computing zeros of functions; as well as, performing a series of other important computing tasks. Learning the programming principles through using LabVIEW better prepares the students for understanding other more complicated text-based programming languages. The virtual instruments developed for several engineering problems are included in this paper to illustrate the special features, capabilities, and utilities of this powerful and interesting software tool.

I. Introduction

In the past several years, a great number of publications have explored various capabilities of LabVIEW relative to laboratory experimentation and data acquisition. Several examples of this type are included in bibliography[1,2,3]. The purpose of the submitted paper is to focus on the programming capabilities of this tool for performing engineering computations and to introduce the reader to some of the potential applications and advantages of LabVIEW.

LabVIEW programs also referred to as Virtual Instruments (VIs) basically have two main windows called the "front panel" and "block diagram". Through using the front panel, the input and the output of the VIs can be managed using a variety of visual controls and indicators available in LabVIEW. All programming steps and constructs used in the development of the solution of problems are included in the block diagram of the VIs as a series of wired graphical icons (nodes). The visual controls and indicators used in the front panel together with the graphical icons of the block diagrams are especially very helpful to the novice engineering students. These visual elements command their attention and interest, and enable them to better understand and grasp the computing and programming, as well as, the engineering principles involved in the development of the solution of problems. The paper does not attempt to teach the reader about all existing possible features of LabVIEW, and nor does it attempt to discuss the details involved in the creating of the LabVIEW program. There are currently several excellent text books[4] available that described the utility of various tools and functions of LabVIEW in great detail.

The LabVIEW tool has a number of attractive features and capabilities that are unavailable in other programming languages. Engineering educators can utilize these tools and capabilities to further enhance their teachings, and promote student learning. The following list of some of the more important and specific advantages of LabVIEW is outlined and discussed below.

(a) Since most people are visual learners, special graphical programming features of LabVIEW are well suited for beginning programmers. Using this powerful software, all important basic programming concepts and constructs can readily be explained to the students in an understandable fashion.

(b) The LabVIEW software provides a powerful, fun, and interesting environment for teaching programming concepts.

(c) Teaching LabVIEW is relatively easy compared to other programming languages.

(d) Various important programming features of LabVIEW can be mastered with ease in a relatively short period. Initial familiarization with some of LabVIEW functions may require a little more time.

(e) LabVIEW has an attractive, convenient, and easy-to-use user interface. Using this interface (LabVIEW's front panel) input data can easily be modified and the resulting changes in the output can instantaneously be viewed. This powerful capability of LabVIEW can serve as a great instructional tool for the faculty and a valuable learning tool for students. In the front panel of the LabVIEW Virtual Instruments, the input and output to the programs can be controlled and displayed using various visual controls and indicators available in LabVIEW. Several different types of controls and indicators are shown in the front panel of the example programs provided in Figures 4, 6, and 12.

(f) Ease with which data from other files such as spreadsheets can be imported and used in the LabVIEW environment.

(g) Ease with which MATLAB script files can be imported and used in LabVIEW. This is a very attractive feature of LabVIEW, since the programs already developed using MATLAB do not have to be rewritten to be utilized. Note that MATLAB is a popular and powerful computing tool which is currently used in many engineering and engineering technology schools around the country and across the world. The utilization of MATLAB in solving engineering problems was discussed in a few earlier publications of the author[5,6]. The capability of importing MATLAB script files into LabVIEW is demonstrated in the block diagram of the truss example problem presented in Figure 13.

(h) Ease with which data can be exported into other file formats to be used with other applications. In the block diagram of the virtual instrument presented in Figure 9 the computed values of stress are exported into an Excel spreadsheet.

(i) Lab VIEW provides great interactive help and documentations for the users to enable them to easily locate and fix their syntax errors.

(j) LabVIEW has great debugging tools. Using a feature of LabVIEW referred to as "Execution highlighting" the movement of data is animated through the motion of a small bubble that moves along the wires in the block diagram from node to node. As the bubble moves through the nodes, the computed values of data are also displayed along the path at the nodes. Using this tool the location of errors within the program can easily be located. LabVIEW also have another valuable tool that can effectively be used to locate logic errors. Using this feature, data passing through any wire in the block diagram can be probed and checked for accuracy.

(k) Flexibility of LabVIEW to handle text-based programming instructions. Using LabVIEW's "formula node" function, the users have also the option of using the text-based programming instructions in their LabVIEW programs whenever more convenient. In the "formula node" mathematical formulas and expressions similar to the C programming language can be used and evaluated. In the block diagram of the example problem presented in Figure 4, a formula node is placed to compute the maximum and minimum stresses in an element of material.

(l) An existing extensive library of virtual instruments in LabVIEW further enhances the functionality of this software tool.

(m) Experience gained from working with basic programming features of LabVIEW will prepare the students to be able to utilize other advanced features of this powerful software tool later in their future studies and careers.

The front panels and the block diagrams for several example problems are included in this paper to further illustrate some of the capabilities and tools of this powerful programming language. In the block diagram of the virtual instruments developed and presented in this paper, as much as possible labels are used to identify various LabVIEW functions utilized. Also, a number additional comments are placed on each diagram to further aid the reader in understanding the basic operations and tasks performed.

II. Stress Analysis

In this section of the paper the development of the LabVIEW virtual instrument for computing the stresses acting on an element of material is presented. Suppose that a square element of material belonging to a solid structural member is subjected to the normal stresses σ_x and σ_y and a shearing stress τ_{xy} as shown in Figure 1. In this problem, the normal stress σ and shear stress τ acting on an inclined surface AB is to be computed together with the maximum and minimum normal and shearing stresses acting on the element.

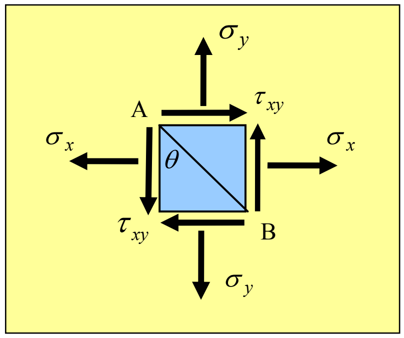

Figure 1. A Sketch of the Element Subjected to Normal and Shear Stresses

The solution for this classical problem is documented in any basic strength of material text[7]. To obtain the stresses on the incline plane shown, the free body diagram presented in Figure 2 is utilized and the force equilibrium equations along the x' and y' directions are written. The obtained equations are shown in Eqs. (1) and (2).

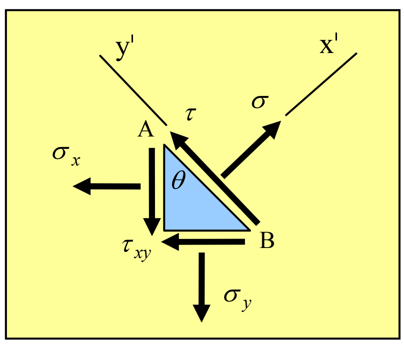

Figure 2. Free Body Diagram of the Wedge

$$\sigma = \frac{\sigma_x + \sigma_y}{2} + \frac{\sigma_x - \sigma_y}{2}\cos(2\theta) + \tau_{xy}\sin(2\theta) \tag{1}$$

$$\tau_{xy} = -\frac{\sigma_x - \sigma_y}{2}\sin(2\theta) + \tau_{xy}\cos(2\theta) \tag{2}$$

Note that normal stress σ and shear stress τ are both functions of θ. To obtain the maximum and minimum values of shear and normal stresses acting on the element, the derivatives of these equations with respect to θ are first set to zero and the obtained values of θ are then plugged back into Eqs. (1) and (2) to yield the maximum and minimum values of normal stress and shear. These stresses are shown below.

$$\sigma_1 = \frac{\sigma_x + \sigma_y}{2} + \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}{}^2} \quad (3)$$

$$\sigma_2 = \frac{\sigma_x + \sigma_y}{2} - \sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}{}^2} \quad (4)$$

$$\tau_1 = +\sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}{}^2} \quad (5)$$

$$\tau_2 = -\sqrt{\left(\frac{\sigma_x - \sigma_y}{2}\right)^2 + \tau_{xy}{}^2} \quad (6)$$

The front panel and the block diagram of the developed virtual instrument for this particular problem are shown in Figures 3 and 4. The block diagram presented in Figure 4 illustrate how the data is obtained from the user and processed through different programming nodes to yield the desired program output. In this diagram, the computation of normal and shear stresses based on application of Eqs. (1) and (2) are performed utilizing the graphical programming language of LabVIEW at the top. The computation of the maximum and minimum values of normal and shear stresses based on application of Eqs. (3)-(6) are performed using a "Formula node" placed at the bottom of the block diagram. In a LabVIEW formula node, text based programming instructions can be issued to perform the computations as demonstrated. Note that in the block diagram in Figure 4 provisions have also been made so that if the allowable maximum value of normal stress or shear stress stated in the problem is exceeded, the color of a failure indicator placed at the bottom of the front panel will turn from a yellow to a red to alert the user.

Note that once a virtual instrument such as the one discussed above is developed, this VI can then be called by another VI to obtain the solution for a different problem. To illustrate this, the front panel and block diagram of a VI developed for computing the normal and shear stresses acting on the inclined surface AB of the bar shown in Figure 5 is presented in Figures 6 and 7. Note that in the block diagram of the problem in Figure 7, the following formula for computing the normal stress is used $\sigma_x = F / A$. In this formula A is the rectangular cross section area of the beam.

To further discuss the capabilities and various other programming features of LabVIEW, in this section of the paper a LabVIEW virtual instrument for plotting the distribution of normal stresses and shearing stresses versus the angle θ is presented and discussed. The block diagram and the front panel for the VI of this problem are shown in Figures 8 and 9. The block diagram of the VI shown in Figure 9 is developed to compute the stresses for a series of values of θ ranging between an initial and a final value using a prescribed interval for θ. This VI is then expected to use the computed values to plot the stress distributions. The tabulated values of stresses and the corresponding plots for a specific set of data are shown in Figure 8.

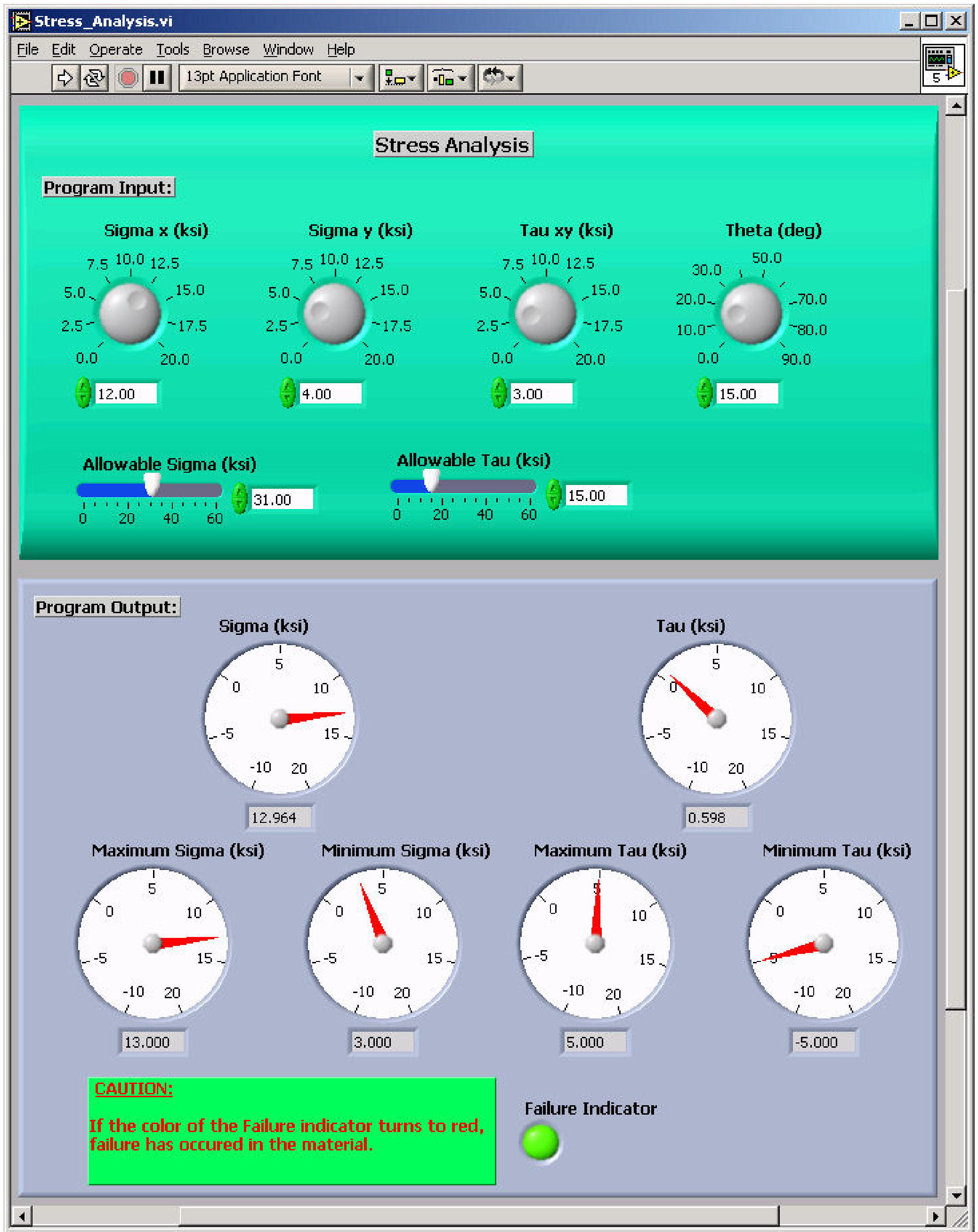

Figure 3. Front Panel of the VI Created for Analysis
of the Stress Acting on an Element of Material

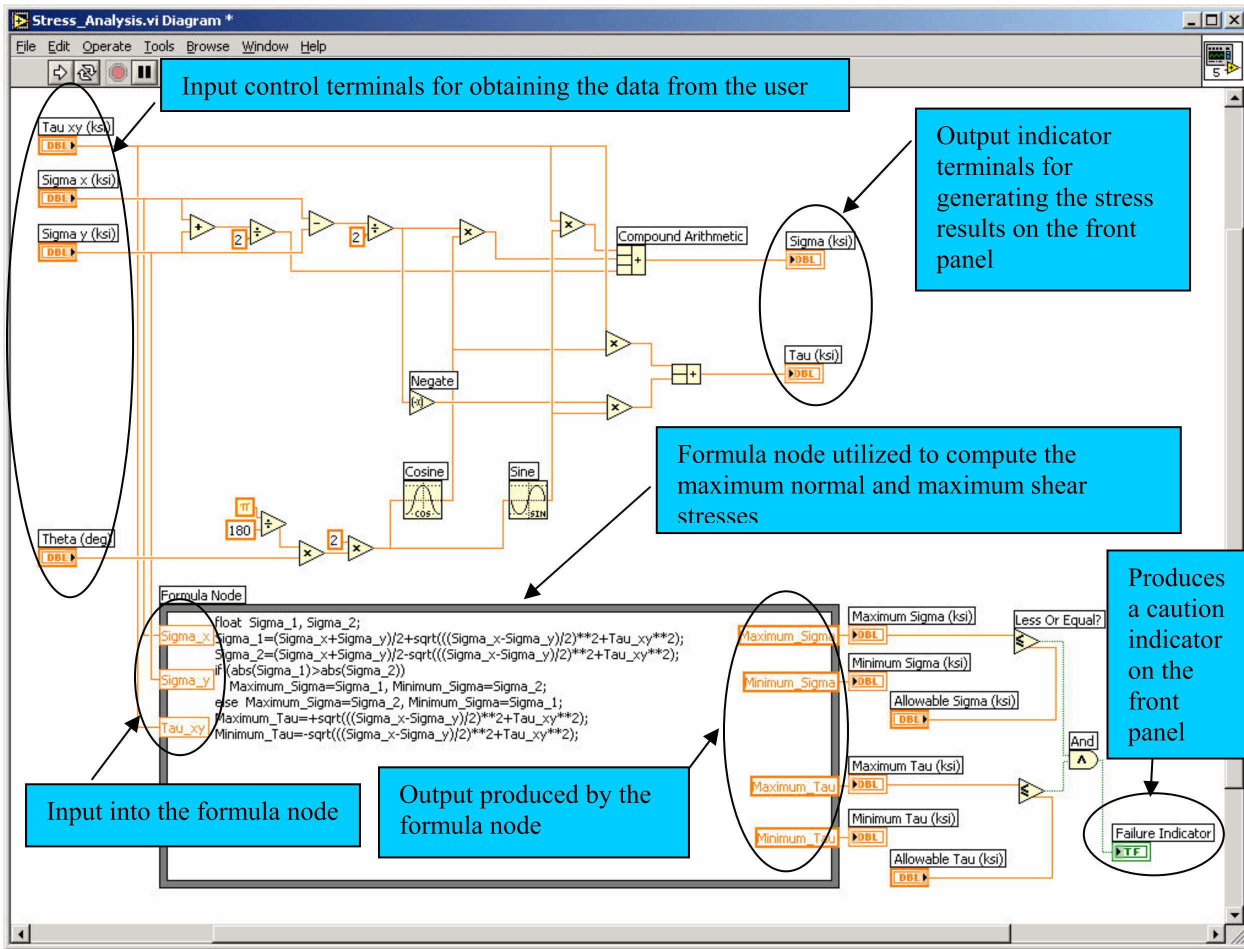

Figure 4. Block Diagram of the VI Created for Analysis of the Stress Acting on an Element of Material

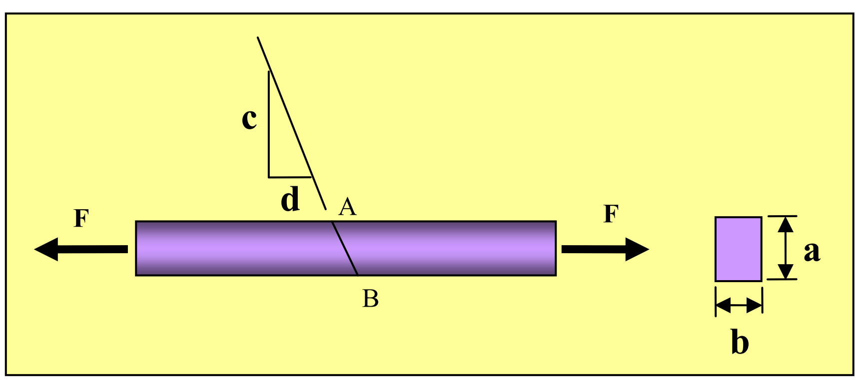

Figure 5. A Bar with a Rectangular Cross Section Subjected to an Axial Force

Note that using the controls placed at the top of the front panel in Figure 8, the user can conveniently change any or all program inputs and interactively view the results produced. The block diagram of this VI presented in Figure 9 illustrates how the data is entered into the VI, processed through using various LabVIEW functions to produce the desired output. Also note that in this diagram provision have also been made to import the results to a spreadsheet file for further analysis.

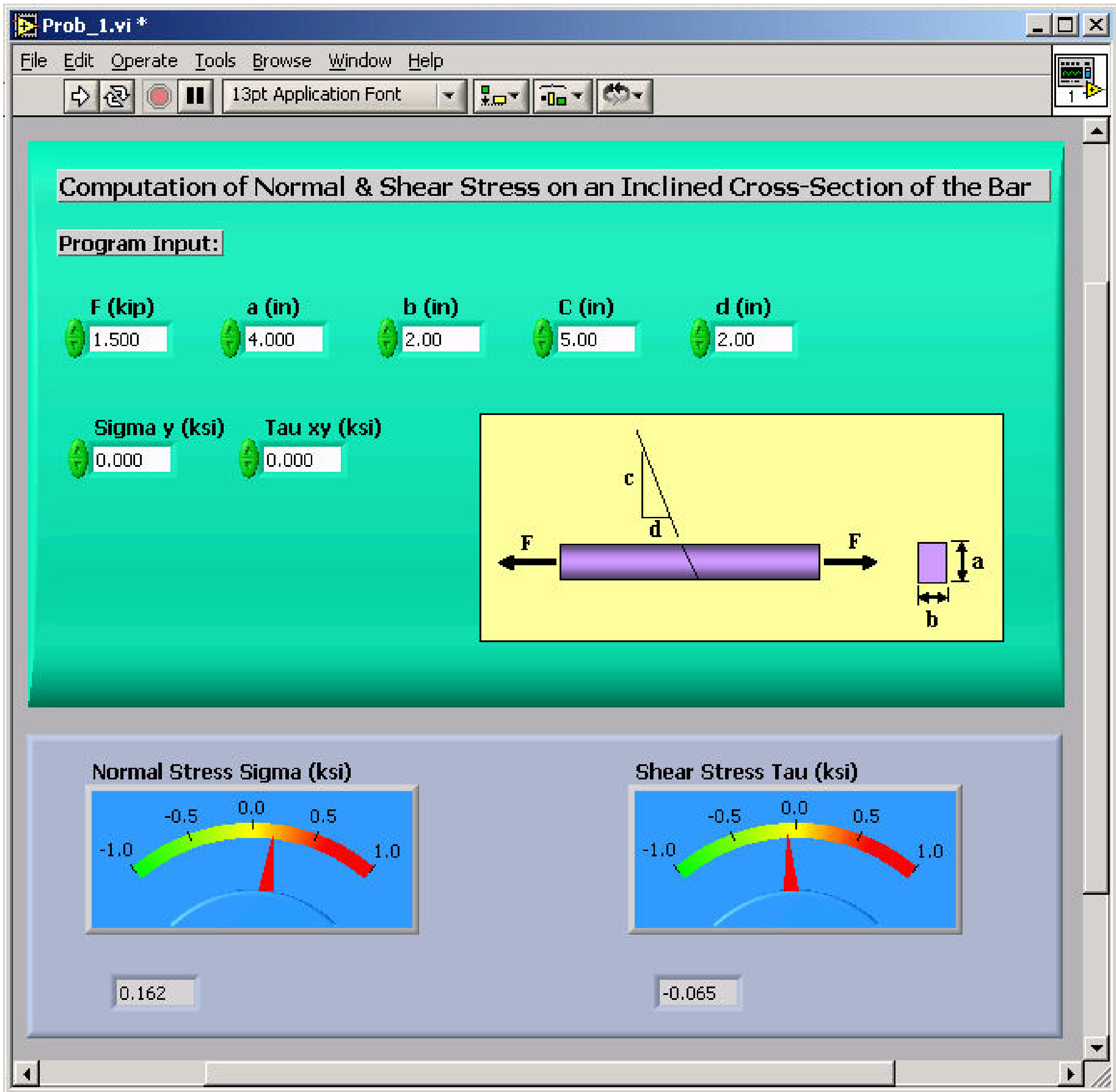

Figure 6. Front Panel of the VI Created for Computing the Normal and Shear Stress Acting on the Inclined Cross Section of the Bar

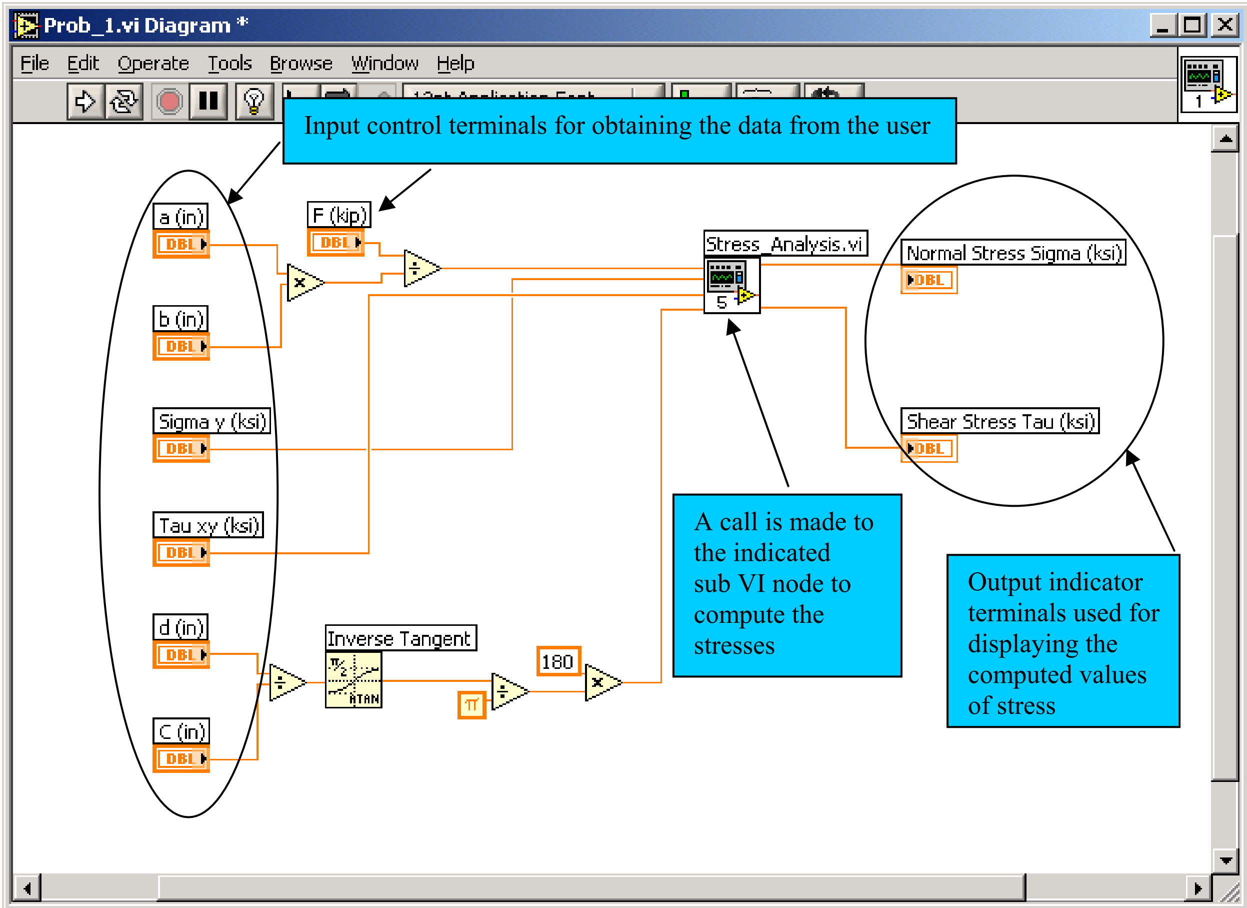

Figure 7. Block Diagram of the VI Created for Computing the Normal and Shear Stresses Acting on the Inclined Cross Section of the Bar

III. Analysis of a Statically Determinate Truss

In this section of the paper, the LabVIEW solution for computing the forces in several members of a truss example[8] using the method of sections is discussed. This example is included to further illustrate the power and capability of this software tool. The block diagram of the VI developed for this problem makes use of a MATLAB script file to perform the actual computations.

The theoretical formulation for determining the forces in this problem using the method sections is outlined first. To compute the forces in members BC, CH, and GH of the truss shown in Figure 10, these members are first cut and the free diagram of the cut section as shown in Figure 11 is considered.

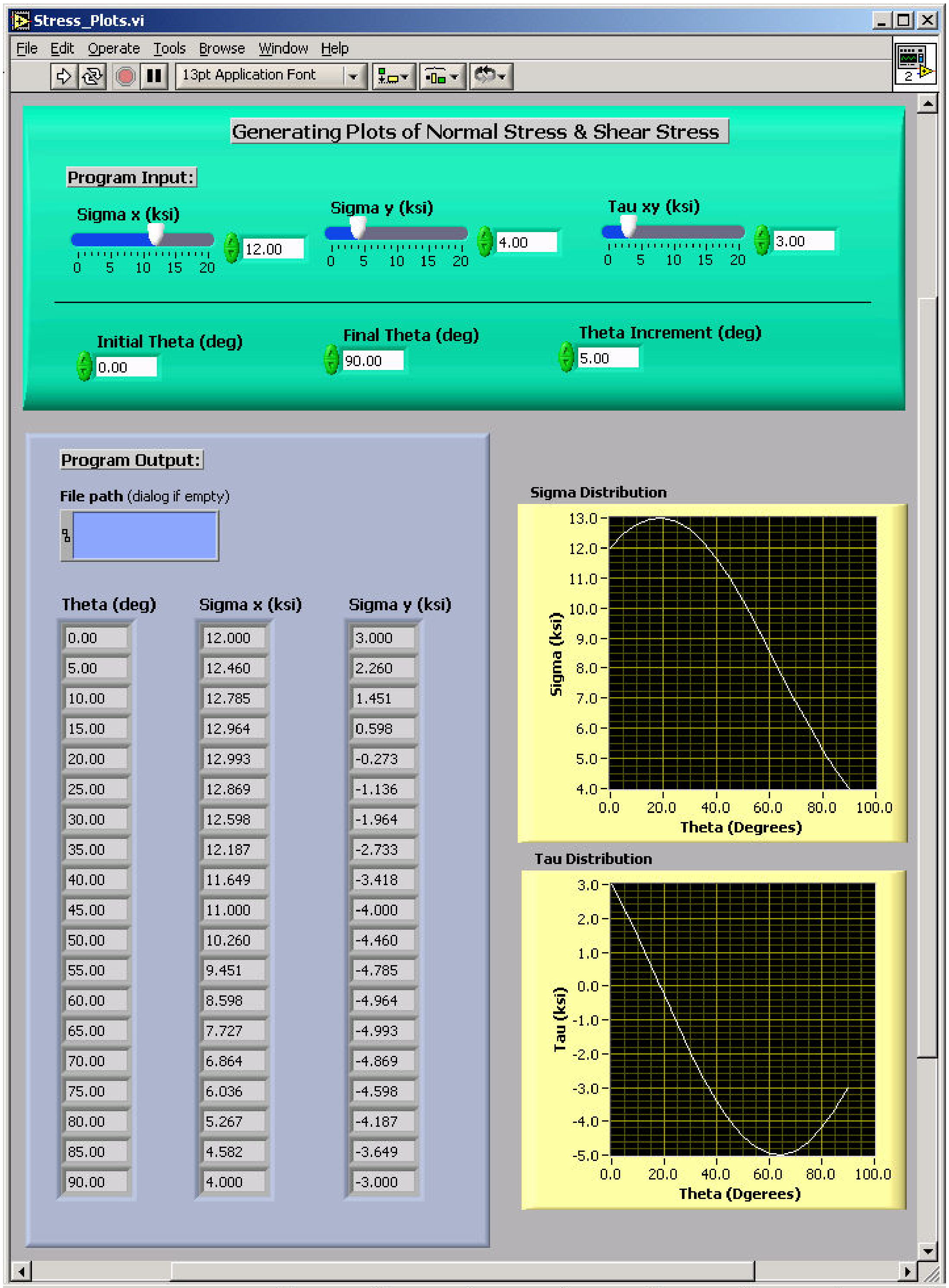

Figure 8. Front Panel of the VI Created for Plotting the
Distributions of Normal & Shear Stresses

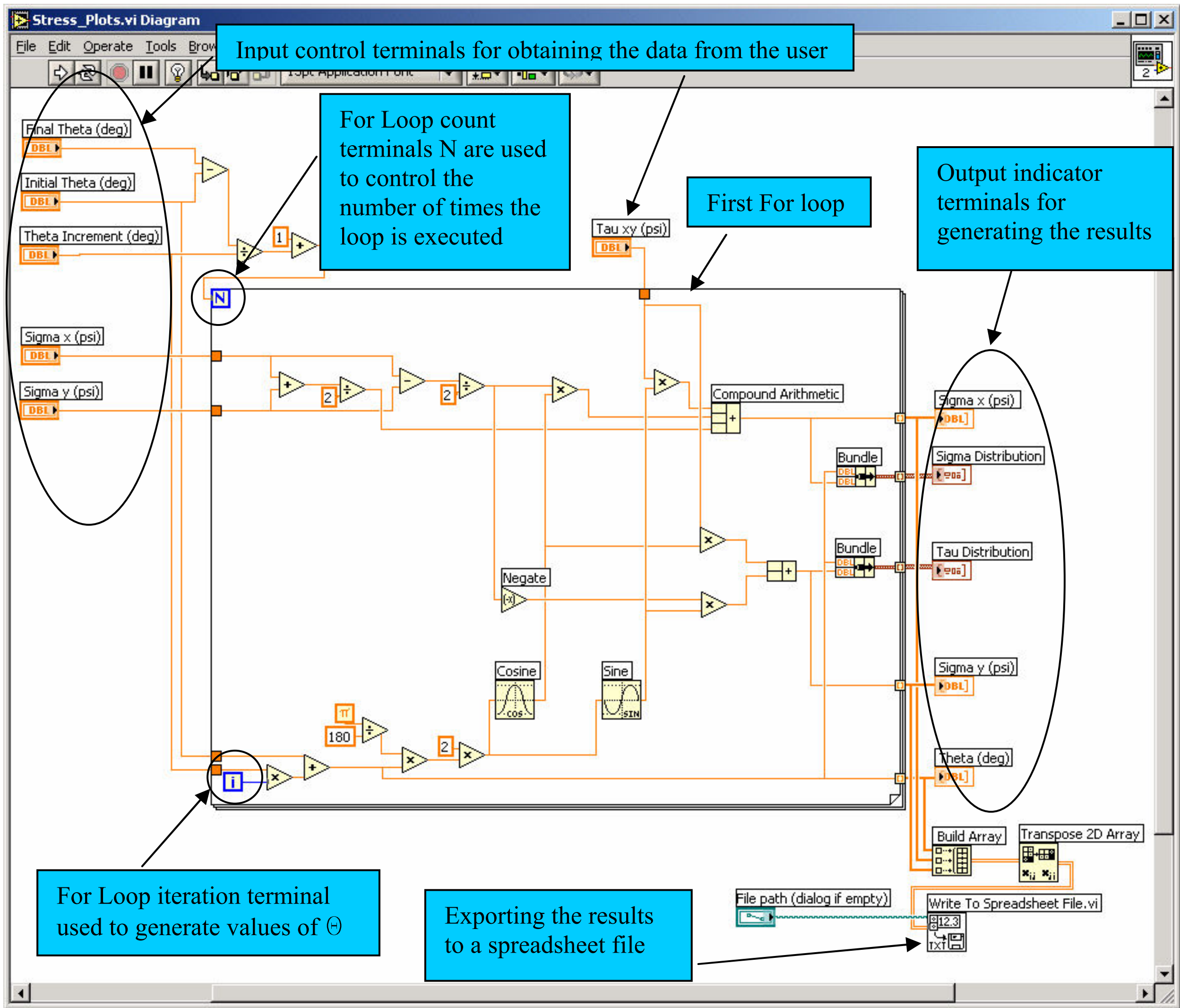

Figure 9. Block Diagram of the VI Created for Plotting the Distributions of Normal & Shear Stresses

The three equilibrium equations of the beam written for the section of the beam shown in Figure 11 can be arranged in the matrix form as show in Eq. 7.

$$\begin{bmatrix} 1 & a/d & a/c \\ 0 & -3b/d & -b/c \\ 0 & -3ab/d & -3ab/c \end{bmatrix} \begin{Bmatrix} F_{BC} \\ F_{CH} \\ F_{GH} \end{Bmatrix} = \begin{Bmatrix} -p \\ -Ay \\ A_y a + 3pb \end{Bmatrix} \qquad (7)$$

Note that in the above equation:

$$A_y = \frac{ra + 2qa - 3pb}{4a}. \tag{8}$$

The equation for A_y (the vertical reaction at support A) is obtained by writing the moment equilibrium equation about point E using the free body diagram of the entire truss. Also note that in Eq. (7), $c = \sqrt{a^2 + b^2}$ and $d = \sqrt{a^2 + 9b^2}$. It should be noted that c and d are the length of members EF and AH respectively as shown in Figure 10.

The front panel and the block diagram for the presented truss problem are shown in Figures 12 and 13. Using the front panel of this VI, the user can enter any specific values for the applied loads p, q, and r, and for the length a and b, and produce the values for the member forces. In the block diagram presented in Figure 13, a MATLAB scripts file is imported to utilize the set of linear equations shown in Eq. (7) to compute the member forces needed. These values are then transferred out of the MATLAB script node to be displayed in the front panel of the VI as illustrated in Figure 13.

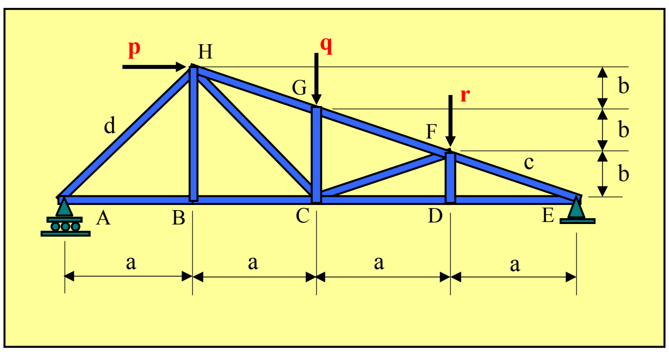

Figure 10. A Statically Determinate Truss

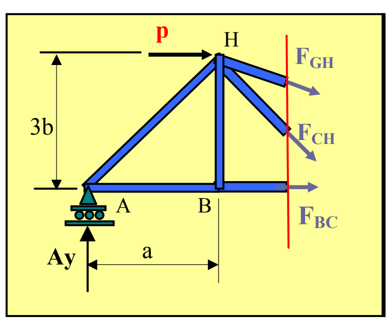

Figure 11. Free Body Diagram of the Left Section of the Truss

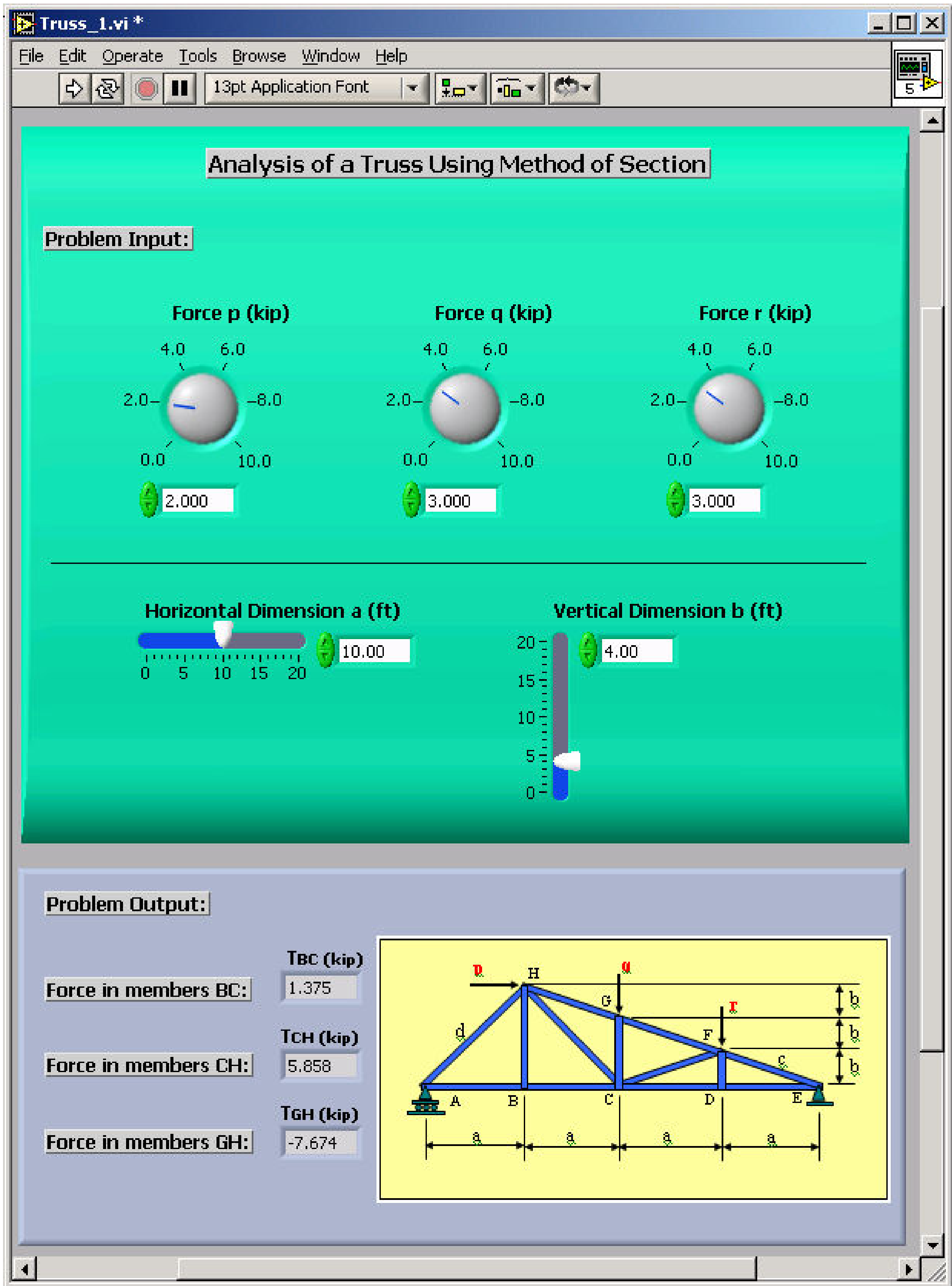

Figure 12. Front Panel of the VI Created for Analyzing a Truss

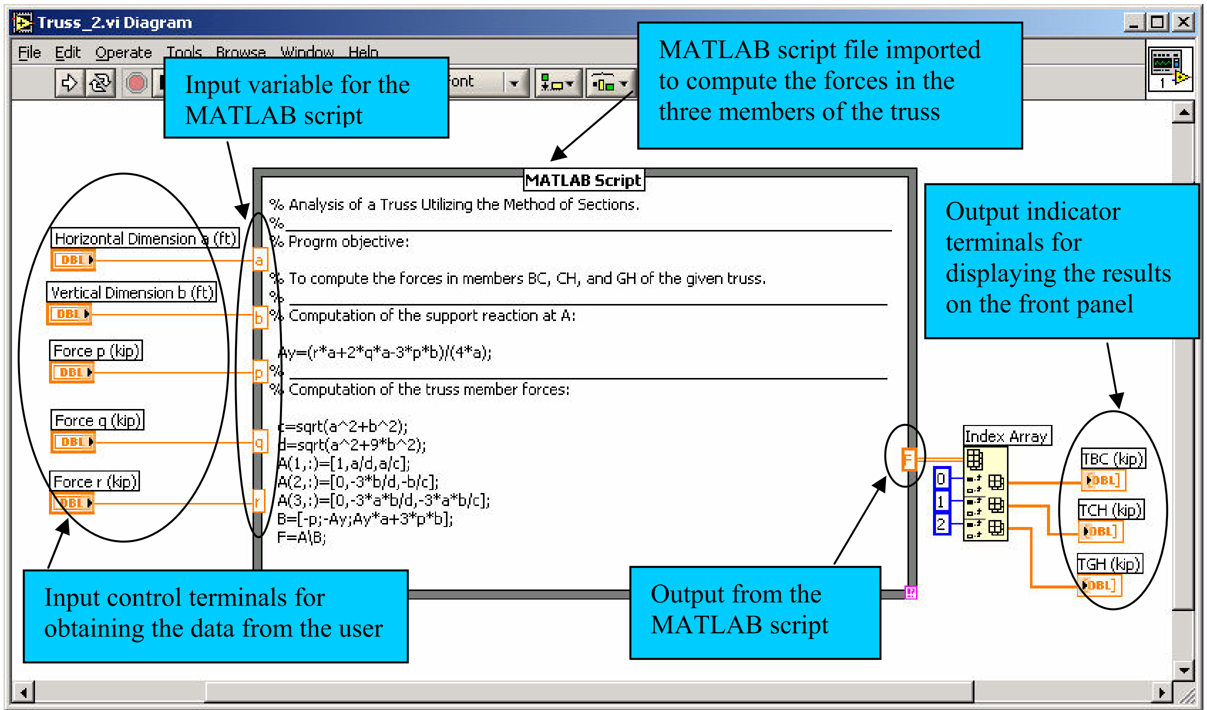

Figure 13. Block Diagram of the VI Created for Analyzing a Truss Utilizing a MATLAB Script

IV. Summary & Conclusion

In the presented paper the use of LabVIEW as a potential programming tool is discussed, and several examples are provided to further illustrate the power and utility of this great software. This programming tool is relatively easy to teach, easy to learn, and easy to use. The software also has many attractive and useful features that are unavailable in other programming languages. A relatively long list of some of the more important advantages and attractive features of this software tool is included and discussed in the introduction section of the paper. At Georgia Southern University, LabVIEW is utilized for the first time this semester to teach a Mechatronics course at the Mechanical Engineering Technology program. The possibility of utilization of LabVIEW in other programming courses at this institution is currently under investigation by the school faculty.

Bibliography
1. Craig, R.R., and McConnell, E.L., "Virtual Instruments Revitalize an Undergraduate Measurements and Instrumentation Course," Proceedings of the ASEE Annual Conference, Charlotte, North Carolina, 1999.
2. Yilmaz, E., "Instrumentation for Relative Cylinder Power Measurement on Internal Combustion Engines," Proceedings of the ASEE Annual Conference, Montreal, Canada, 2002.
3. Bachnak, R., Steidley, C., "Data Acquisition for Process Monitoring and Control," Proceedings of the ASEE Annual Conference, Nashville, Tennessee, 2003.
4. Bishop, H. B., Student Edition LabVIEW 6i, Prentice Hall, 2001.
5. Navaee, S., Das, N.K., "Utilization of MATLAB in Structural Analysis," Proceedings of the ASEE Annual Conference, Montreal, Canada, 2002.
6. Navaee, S., "Developing Instructional Modules for Analyzing Structures," Proceedings of the ASEE Annual Conference, Nashville, Tennessee, 2003.
7. Bauld, R. B., Junior, Mechanics of Materials, Second Edition, PWS Publishers, 1986.
8. Hibbeler, R.C., Structural Analysis, Third Edition, Prentice Hall, 1995.

SHAHNAM NAVAEE
Shahnam Navaee is currently an Associate Professor in the Engineering Studies Program at Georgia Southern University where his primary responsibility is teaching freshman and sophomore level courses. Dr. Navaee received his B.S. and M.S. degrees in Civil Engineering from Louisiana State University in 1980 and 1983 and his Ph.D. degree from the Department of Civil Engineering at Clemson University in 1989.