

Inclusion of Hands-on Interactive Programs for Teaching Statics

G.V. Narayanan

University of Toledo
nara@utoledo.edu

The Statics course is the first course in mechanics that is offered to the sophomore/junior mechanical engineering technology student. It is always a challenge to the student in terms of good understanding of the subject. Of course, the instructor plays an important part for his better understanding of the subject, even if the textbook is very good. The instructor knows the consequences of giving too many homework problems for each concept. Giving too few problems pose situations in which the students do not understand or master the subject matter. With the increase of students' ability to use computers and programs, the instructor can introduce hands-on interactive programs that aid in teaching statics. There are not many such programs available specialized for teaching statics. Many commercial programs do exist for performing structural analysis, but not good enough to teach the principles of statics at a slower pace. In this paper, the hands-on interactive programs or modules, created or developed so far, are discussed in terms of teaching basic mechanics and statics. Such mini-programs can be interesting and rewarding to students in doing homework problems. Also, the student can do many more statics problems with the help of these programs than without them. It is also rewarding for the instructor who can make students learn more from this hands-on interactive approach.

Introduction

The Statics course is the first course in mechanics that is offered to the sophomore/junior Mechanical Engineering Technology (MET) student. It is somewhat of a difficult subject for the student who comes across mechanics for first time. The student is challenged with the concepts and understanding of the subject. A good instructor can help in the understanding of statics so the student is proficient in solving a variety of statics problems. One of the known teaching methods of making the student understand the subject is letting the student do a variety of statics problems. This will not only increase the student's understanding of statics concepts and finding its engineering solution, but also this approach will let the student experience and gain knowledge of a variety of statics problems occurring in common engineering systems.

There exist many good texts that will aid the student in terms of experiencing and gaining the knowledge of a variety of statics problems. However, due to the students' weakness in general mathematical analysis, many ET students have difficulty in reading these good texts as a self study. Even with the good instructor, the student cannot learn this subject unless the student himself experiences and does solve a good many statics problems.

However, there is not sufficient course teaching time in a semester for the instructor to do a handful of problems associated each concept. Thus, this adds up to a total of 65 to 70 problems for the student to do in a semester, even with a minimal number of problems in each concept.

This poses a challenge for the student in terms of time and energy to solving that many problems on their own, and in terms of doing a good job to understand the subject. To alleviate some pressure off the student's time in terms of numerical crunching and manual doing, some special computer programs can be created by the instructor for helping the student in reducing time and increasing understanding. This also helps the instructor in minimizing with his class teaching time on repetitive number crunching work in class on many problems.

The creation of such statics programs for the purpose of helping to teach statics takes some effort, and hence, such creation cannot be suggested during the course development time. Also, such statics software does not exist on the store shelf that will help the instructor and students in this direction. This paper attempts to create such programs as a help in teaching future statics students. In this paper, a set of such statics programs are identified and a preliminary attempt to set its graphical user interface for such mini-programs, where possible. These programs are developed using Visual Basic .Net and MATLAB, as per availability and easiness possible on the students' understanding and associated learning environment. The programs are not in its final form, but this paper is an attempt to start the discussion among other statics instructors.

Mini-Programs that aid in teaching Statics

The following teaching mini-programs are designed that will help the ET student in an understanding and solving of statics problems. It is to be noted that this is such a first attempt by the author. The break down follows the subject order of teaching statics.

- 1) Resultant of Two Forces Program
- 2) Force Resolution Program
- 3) Dot Product Program
- 4) Cross Product Program
- 5) Solutions of Equations Program
- 6) Equilibrium of Forces at a Point
- 7) Computing Reaction Forces on 2D bodies
- 8) Computing Reaction Forces on 3D bodies
- 9) Force Analysis of Determinate Truss/Frame Structure

Resultant of Two Forces Program

This Visual Basic .Net program, shown in Figure 1, is designed so the student has flexibility of input of two forces. The student must be able to enter each force either in magnitude and direction angle form or in vector component form. At a minimum, the program will compute the resultant, and display the forces in both forms. However, the

program is designed to let the student experiment with various input. This program flexibility will entice the student to work with the program to enter many different force values, and in the process, the student will build not only confidence of how to compute the resultant, but will better understand how the resultant of two forces is calculated. At student's option, the program will print intermediate steps involved in computing the resultant. An option to plot the forces will be given. An example with two force input values is shown in Figure 1. The intermediate prints and plots are not available today.

Resolution of a Force into Two Directional Forces Program

This Visual Basic .Net program is an extension of the previous mini-program with an option to resolve a force into two given directions. The student must give two directions for a given force to be resolved in these directions. The student must reset the screen so that he is not confused with previous computed values. The student inputs 'Force R' and directions of 'Force A' and 'Force B'. This program, shown in Figure 1, will be demonstrated at the conference, and it is difficult to show the interactive usefulness of this program on a technical paper.

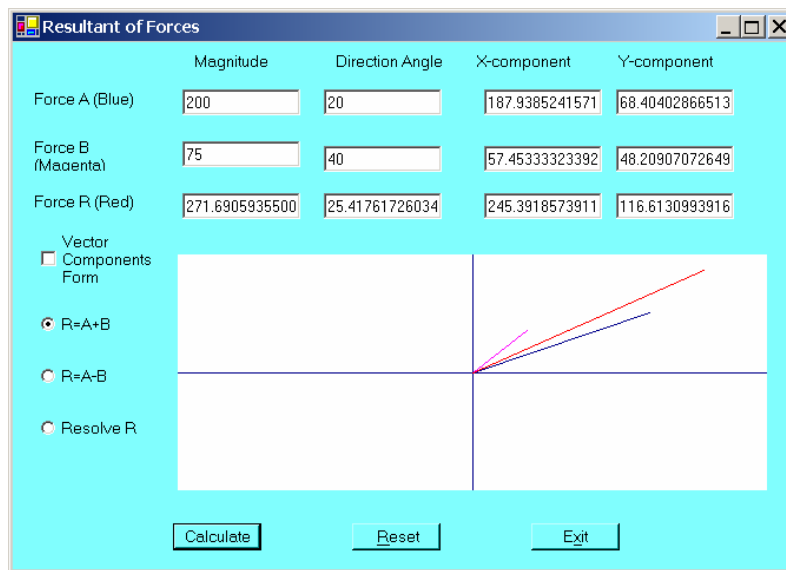


Figure 1: GUI for Resultant of Two Forces

Dot Product Program

The projection of one force onto a direction of the other force is somewhat of a difficult concept to the student in terms of computing a numerical number. It is easy for the student to visualize such a projection, but when asked to compute its projection in numerical form, the student gets stumped. This MATLAB GUI program, shown in Figure 2, can compute the projection, and it will display the intermediate calculations so the student has better appreciation of the projection of one force on to the other. Such intermediate calculations printing GUI option is yet to be developed for this mini-program.

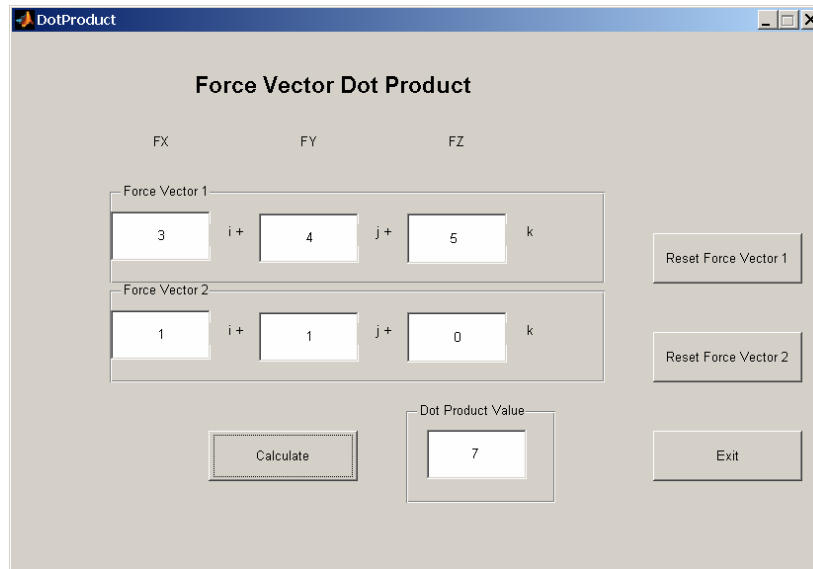


Figure 2: GUI for Dot Product Program

Cross Product Program

The creation of a vector direction perpendicular to two given directions is a very difficult concept for the statics student. Even if the cross-product in component form is discussed, the student finds it difficult to understand its computation unless the student does it more than a couple of times by himself. This MATLAB GUI program, as shown in Figure 3, is designed to help in that direction. An attempt is being made to have the program display how the cross-product components are computed. This simple display may help the student to digest the cross-product computation to compute the perpendicular vector. A GUI option to compute the unit vector of a perpendicular direction will exist in the future version of this mini-program for the student. This option will help the student understand as to how a unit perpendicular vector can be calculated from two vectors in three dimensions. This important cross product concept is used in many engineering applications and commercial applications. It is an important concept that the student must be comfortable with in many engineering subject areas.

Solution of Equations Program

In solving for statics equilibrium equations in two or three dimensions, the student often needs to solve simultaneous equations in two or three variables. The student calculator has the ability to solve such equations-solution problems. However, most calculators do not teach the student how these equations are solved. Many students have trouble in solving simultaneous equations with two variables. The MATLAB statics program is designed as a script, as shown in Figure 4, to help the student understanding in this direction, in addition to solving for numerical numbers. A symbolic expression solution will be attempted in future with the use of MAPLE facility within MATLAB. Such a

symbolic development is somewhat more difficult, to be done in a general form, with the student understanding in mind, and hence, it is not attempted at this time.

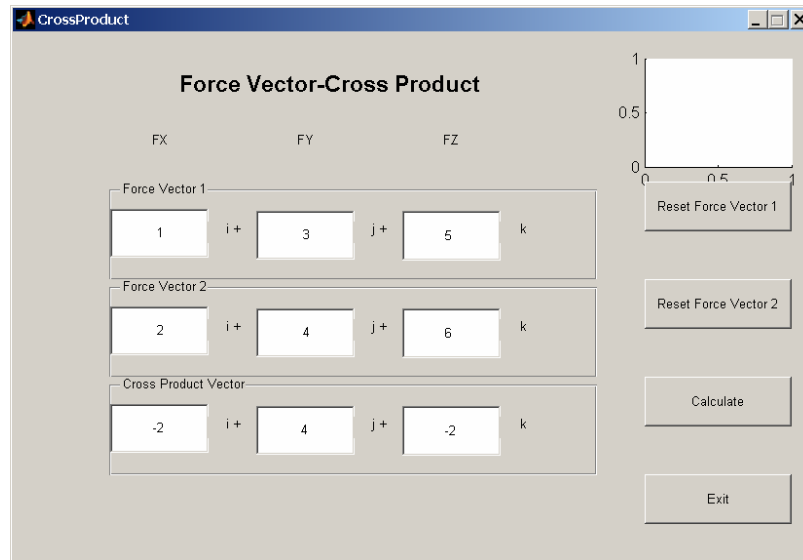


Figure 3: GUI for Cross Product Program

```
% MATLAB Script to solve Simultaneous Equations
disp('Program to solve Equations: am*xv=bv');
disp('Enter LHS coefficients as a matrix, denoted by
am');
disp('Example LHS input: [2.1 3.2 3.4;4.5 5.4 6.8;45.
34. 32]');
am = input('Enter LHS coefficients as a matrix:');
disp('Enter RHS as a column vector, denoted by bv');
disp('Example RHS input: [45.2; 64.8; 72.4]');
bv = input('Enter Right Hand Side vector:');
disp('LHS matrix and RHS vector entered:'), am, bv
disp('The solution vector, denoted by variable xv:');
xv = am\bv
clear;
```

Figure 4: MATLAB Script to solve simultaneous Equations

Equilibrium of Forces at a Point

The use of MATLAB script program helps the student after he formulates the force equilibrium equations. The forces can act at a point either on a plane as in 2D equilibrium or in space as in 3D equilibrium. The student gets the practice of formulating the equilibrium equations at a point, on paper. The student can then solve equations using the MATLAB script in Figure 4 that solves simultaneous equations. But two additional MATLAB scripts, shown in Figure 5 and Figure 6, are given below that allow

the student to enter 2 or 3 equations directly, in 'symbolic variables' form, as in 2D or 3D problems, respectively. An equivalent GUI program can be developed that includes this script inside the GUI program. However, the student will not learn as to how the solution is obtained. Since the student learning is important, it is suggested in this paper to let the student enter this script on a file and use it in a MATLAB session as a script command. This little extra work will help the student indirectly to learn the programming aspect within MATLAB for solving the equilibrium equations.

```
% Solve Equations of Motion in 2-D problems
disp('Use symbols F1 and F2 for two unknown forces');
disp('X-direction force equilibrium equation');
disp('example input: 3.6 * F1 + 4.2 * F2 = 58.4');
eqn1c=inputdlg('Enter sum (Fx-comp) force equation:');
disp('Equation 1 entered:'), eqn1c
disp('Y-direction force equilibrium equation');
eqn2c=inputdlg('Enter sum (Fy-comp) force equation:');
disp('Equation 2 entered:'), eqn2c
eqn1=char(eqn1c);
eqn2=char(eqn2c);
disp('Solution for two unknown forces')
[F1,F2]=solve(eqn1,eqn2);
disp('Force F1: '), F1 = double(F1)
disp('Force F2: '), F2 = double(F2)
clear;
```

Figure 5: MATLAB Script to solve 2-D Equations of Motion

```
% Solve Equations of Motion in 3-D problems
disp('Use symbols F1, F2, F3 for three unknown
forces');
disp('X-direction force equilibrium equation');
disp('example input: 3.6 * F1 + 4.2 * F2 - 6.3 * F3 =
58.4');
eqn1c=inputdlg('Enter sum (Fx-comp) force equation:');
disp('Y-direction force equilibrium equation');
eqn2c=inputdlg('Enter sum (Fy-comp) force equation:');
disp('Z-direction force equilibrium equation');
eqn3c=inputdlg('Enter sum (Fz-comp) force equation:');
disp('Entered equations are:'), eqn1c,eqn2c,eqn3c
eqn1=char(eqn1c);
eqn2=char(eqn2c);
eqn3=char(eqn3c);
disp('Solution for three unknown forces');
[F1,F2,F3]=solve(eqn1,eqn2,eqn3);
disp('Force F1: '), F1 = double(F1)
disp('Force F2: '), F2 = double(F2)
disp('Force F3: '), F3 = double(F3)
clear;
```

Figure 6: MATLAB Script to solve 3-D Equations of Motion

Computing Reaction Forces on 2D Bodies

The statics problems of forces acting on bodies are much harder for the ET student, and it would be prudent to develop knowledge based programs to help the student in this understanding of solving these kinds of problems. However, such knowledge based program development is outside the scope of this paper. In this paper, the computation of reactions for forces on 2D bodies is suggested to be done using the MATLAB scripts in Figure 4 and 6. The problem here is broken into two phases, first that of formulation of equations on paper, after drawing the free body diagram, and then using the appropriate MATLAB script to solve for the reactions. Of course, the script in Figure 4 can be used for any number of unknowns, even though it is discussed as a matrix equation with three unknown forces. On the other hand, the script in Figure 6 can only be used only with 3 equations with 3 unknown forces.

Computing Reaction Forces on 3-D Bodies

The script in Figure 4 can be used for solving 6 unknown forces in 3D equilibrium equations of motion. The responsibility of drawing the free body diagram and hence formulating the equations lies with the student. The script in Figure 4 can only help with the solution of simultaneous equations. A script similar to Figure 6 for 6 unknown forces can be developed, but this has not been done in this paper.

Force Analysis of Determinate Truss/Frame Structure

There exist many commercial finite element programs that can perform displacement analysis of trusses and frames, both determinate and indeterminate structures. However, in statics, the student needs to be introduced to the force analysis of determinate truss or frame structures only. An added difficulty of finding programs that will teach the student such a force analysis understanding of trusses and frames confirms the developmental need of force-analysis-teaching program applicable for determinate trusses and frames. In this paper, only a preliminary MATLAB GUI interface is developed as a teaching program for analyzing determinate trusses and frames. The implementation and testing of such program is underway, and is expected to be completed next year. Hence, no interface figure is available at this time for this paper. In the interim, the MATLAB script in Figure 5 can be used for 2D truss member force analysis using the method of joints. The student will use Figure 5 script for each joint on the truss. Similarly, the MATLAB script in Figure 6 can be used for 3D truss member force analysis using the method of joints.

Classroom Use of Statics Mini-programs

The statics mini-programs have been offered in the MET Statics course during the 2004 Fall semester, as a set of optional virtual laboratory exercises. No special computer laboratory sessions were used to present the mini-program materials. The class room lecture materials were presented according to schedule, with the additional mini-

programs introduced as options for students to use to do their homework problems. Neither any special effort was taken to measure the success of this teaching approach, nor any measurement done in class during testing to see if the student liked these mini-programs. Such measurements may be done in the near future after these mini-programs are complete with instructions and intermediate printouts to help ET students in the understanding of solving engineering statics problems.

Conclusions

In this paper, a set of independent mini-programs are discussed that help the ET student in understanding the subject of engineering statics. Even though some preliminary testing of many mini-programs has been done, these programs are not ready for public distribution. These programs use either Visual Basic .Net or MATLAB environment for its development. Such a program development forms the first phase of having a statics teaching tool before a knowledge-based software will be attempted for teaching statics. If the knowledge-based software is successful and possible in statics, the knowledge-based software development process will be extended to help create a teaching tool for dynamics.

References

- 1) Hibbeler, R.C., "Engineering Mechanics: Statics", Tenth Edition, Pearson Prentice Hall, 2004
- 2) Chapman, S. J., "MATLAB Programming for Engineers", Second Edition, Brooks/Cole, 2002
- 3) Deitel, H. M., Deitel, P. J., and Nieto, T. R., "Visual Basic .Net – How To Program, second Edition", Prentice Hall, 2002

Dr. G. V. Narayanan teaches at the University of Toledo, Toledo, Ohio. He can be contacted by email at 'nara@utoledo.edu'.