

Comparative analysis of spreadsheet and MATLAB in beam bending calculations

Jacek Uziak & Jerekias Gandure

University of Botswana
Gaborone, Botswana

ABSTRACT: Strength of materials is one of the fundamental courses for any engineering undergraduate programme. Knowledge in that area is essential for any design and construction as engineers need to understand material strains and stresses used for particular applications. In that respect the theory of beams is indispensable in modelling mechanical and civil engineering problems. In this article, the authors compare the attributes of a spreadsheet and a MATLAB script as an educational tool for undergraduate students. Both are used to solve bending of beam problems. Attributes of interest include interface friendliness, programming difficulty, speed of execution, accuracy of output and cost of acquiring software. The study revealed that the two software packages are equally accurate. It also revealed that although MATLAB is faster and has a better user interface than the spreadsheet, its use by learners is limited by its high purchase price.

INTRODUCTION

The study of strength of materials is critical for any engineering student. Knowledge in that area is essential, especially in design and construction of any types of structures. The study of strains and stresses enable appropriate choices of suitable materials for any application [1].

The theory of beams is indispensable and is a basic part of the strength of materials. Beam theory makes it possible to model problems of mechanical and civil structures like bridges, boats, vehicles and others. In teaching, it is important to couple beam theory with a computational tool to improve the quality of students' learning. In this article, the authors compare MATLAB and spreadsheet calculations in solving beam bending problems. Both are used as an educational tool giving an opportunity to obtain results for complicated problems without tedious mathematical calculations. Students can use the results obtained from the software to validate the correctness of their work during individual study, thus, improving their ability to work independently without supervision.

This work seeks to achieve two broad objectives. Firstly, it seeks to develop an educational tool for students with two different software packages, MATLAB and Microsoft Excel, and compare correctness of output to assess justification for the cost of acquiring MATLAB against use of Excel which is readily available. Secondly, the work seeks to ensure that the developed educational tool for students has the capability to deal with a maximum range of beam problem cases, and to evaluate the most important forces within the beam, like the shear force, the bending moment and deflection. It is essential to study responses of material to external loads.

In order to achieve the stated objectives, some knowledge of beam theory is necessary in order to develop programs. Methods adopted for program construction in both MATLAB and Excel are also important and are briefly explained in this work. Finally, after getting results from both programs, analyses of differences, constraints and limitations of MATLAB and Excel programming, and a comparison between the two programs were performed.

BEAM THEORY

Beams are structural elements used in the construction industry and most of *strength of materials* problems are based on the concept of beams. A beam carrying several loads allows a real strength of materials problem to be simulated [2]. A structure where forces are determined by considering only its static equilibrium is said to be statically determined. But, in practice, forces in a structure cannot be obtained by considerations of static equilibrium alone. External mechanical loads create some internal forces [3]. When the structure is not subjected to any loads, these forces cancel themselves out according to the principle of reciprocal action. But, when a beam carries lateral forces, two important types of loading action are set up at any section, and these are bending moment and shear force. Strains are the

consequence of internal forces, allowing for equilibrium of the system. In order to quantify internal forces, a mathematical tool called stress is used. Associated with stress are two internal actions created by the effect of lateral forces; namely, bending moments and shear forces.

SOLUTION METHODS

MATLAB Solution for Beam Problem

MATLAB is interactive software based on matrix computation (MATrix LABoratory) [4][5]. It allows high-performance numeric computation and visualisation. Numerical analysis, programming and graphics are integrated in this software. The extraordinary feature of MATLAB is that there is no distinction among real, complex and integer numbers. Any variable can take any type of number without special declaration in programming. This makes programming faster and more productive. This is enhanced by many incorporated functions and toolboxes, aiding the precision of results. Another very important tool in MATLAB is Graphics. Graphic presentation is possible with just a few commands, and scientific and artistic graphic objects can be created on the screen using mathematical expressions. MATLAB solutions for beam problems are accomplished in steps; namely, choice of method, command window, M-files and process.

The first step in building a MATLAB program is to select a method. This is followed by establishing expressions for beam functions for a particular case and writing appropriate subroutines. The *Command* window provides the primary place of interaction with MATLAB. The data and variables created in the *Command* window reside in the MATLAB workspace. For simple problems, entering requests at the MATLAB prompt in the Command window is fast and efficient. However, as the number of commands increases or in the case when the value of one or more variables has to be changed, typing at the MATLAB prompt become tedious. MATLAB provides a solution called M-file, which allows MATLAB commands to be placed in a simple text file that can be accessed by the software to evaluate commands as if they were typed at the MATLAB prompt as shown in Figure 1.

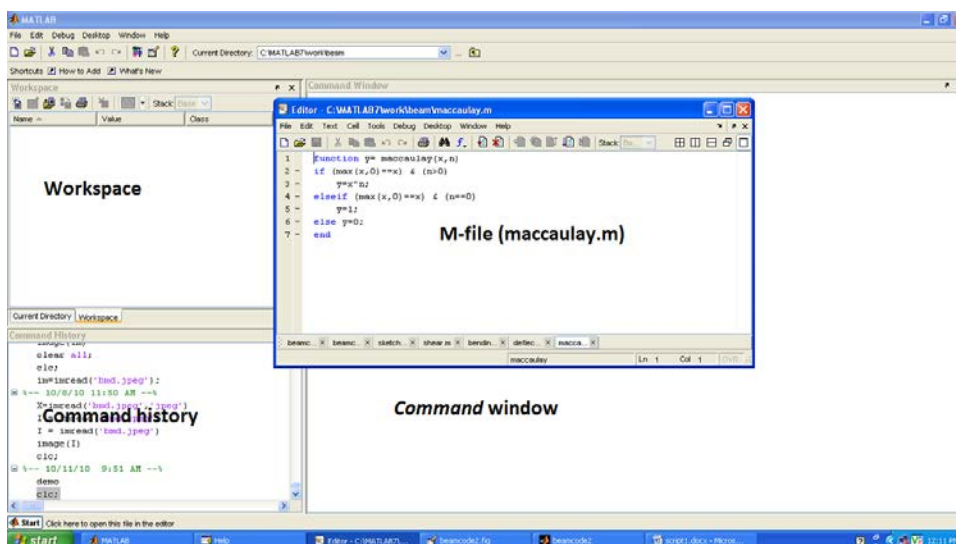


Figure 1: MATLAB window.

A new M-file should be created in order to introduce the singularity functions. Thereafter, they can even be written in the command window or other M-files to act as functions as shown in Figure 2. Thus, every function, which is used regularly must be written in an M-file.

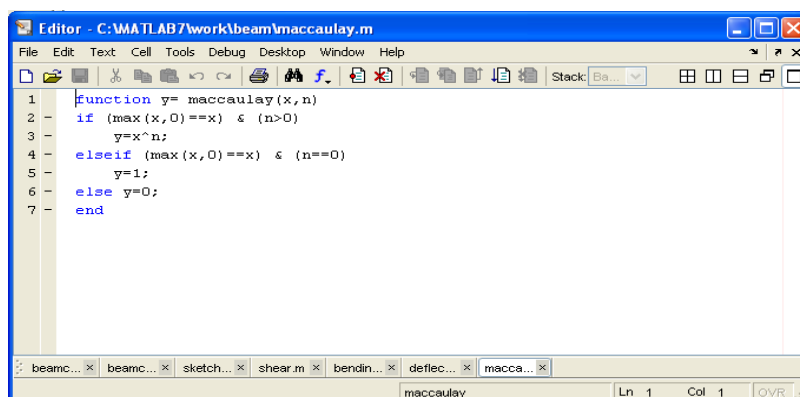


Figure 2: Macaulay's function.

For every beam, the process of evaluation is based on the most general problem where all typical loads are applied. Users can choose between a cantilever beam, simply supported beam or a fixed beam. A dialog box will, then, appear with all the relevant data. By default, all data are equal to zero at the start of the program, but according to a user's problem, values of loads, coordinates of forces and moments can be adjusted.

The programming in MATLAB for the beam bending problem is, indeed, straightforward for students familiar with basic programming and the MATLAB environment. Special care has to be taken to deal with non-linear terms in cases where a Macaulay's method is used. In a MATLAB solution, the continuous functions for the shear force, bending moment and deflection as given by the fundamental bending equations are computed at discrete points on the beam. The values computed at the discrete points give the exact values of the functions at these points. However, the smoothness of the curves to be plotted for these functions, depend on how close the discrete points are made to each other [6]. It is clear that very smooth curves can be obtained by discretising the beam into a large number of segments. However, if that number is very large, the number of computations to be done by the computer on all the equations will also be very large and the computation time can be quite long.

Spreadsheet Solution for Beam Problems

Microsoft Excel is a *number crunching* application, which is mostly useful for the accountants, though it also provides scientists and engineers with a powerful computational tool. Microsoft Excel is widely available (MS Office) and is easy to use. Once a few basic skills have been mastered, many spreadsheets may be developed in much the same way one would precede with pencil, paper and calculator, but with higher speed, precision and flexibility. Microsoft Excel has a range of functions that can be utilised for the analysis and presentation of results. Understanding Excel spreadsheets begins with knowledge of worksheet components that include toolbars, formula bar, sheet tabs, cells and cell ranges. Figure 3 shows a worksheet in Excel.

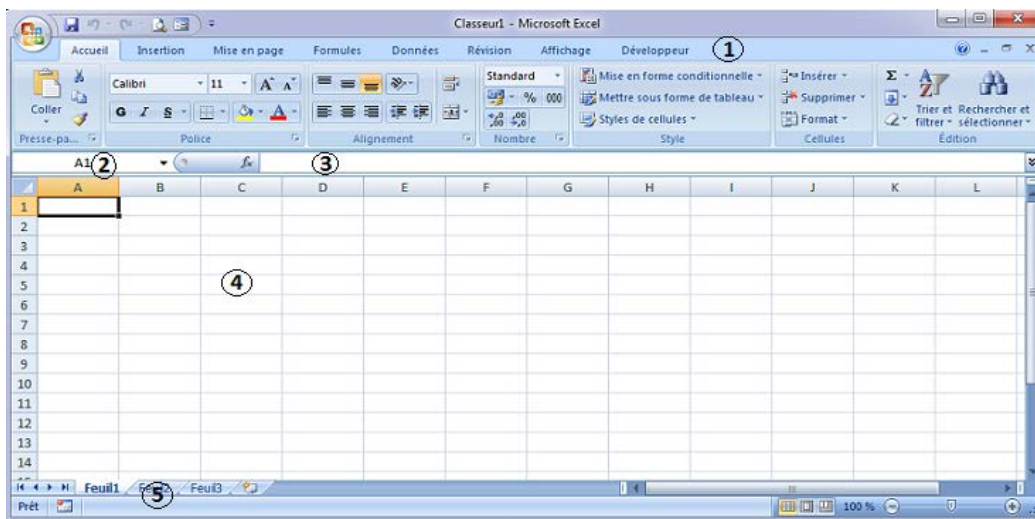


Figure 3: Excel worksheet.

In order to improve interactivity in the Excel software, there is a useful tool called Macro. A macro is a set of instructions that tells Microsoft Excel to perform one or more actions for programmers. Macros are like computer programs, but they run completely within Excel. They can be used to automate tedious or frequently repeated tasks. The overall process for recording a macro consists of three steps. Firstly, a macro recorder is started, Figure 4, and a name for the macro supplied. Next, actions seeking to be recorded are performed, such as choosing menu commands, selecting cells or entering data. Finally, the macro is saved and recorder stopped. The macro system is activated at the point of launching Excel worksheet.

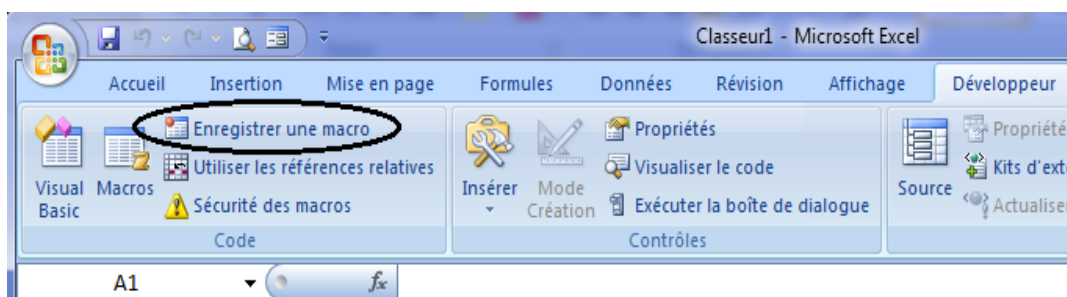


Figure 4: Macro recorder.

Excel spreadsheets use a solver to compute unknown values. The solver can handle problems that involve many variables and cells, and can help the user to find combinations of variables that maximise or minimise a target function. It also specifies constraints that must be met for the solution to be valid. Its ability to manage a large number of constraints makes it very useful in solving beam problems, which normally involve systems of equations. The spreadsheet provides a *help* function for troubleshooting during program construction and operation.

The principle of spreadsheets is to drag and, therefore, repeat a calculation on a point to a very large number of points. To be able to take advantage of this feature, it is necessary to proceed with closed form equations. That is why the energy method for beam calculations cannot be used in Excel.

The beam bending spreadsheet can be applied to both statically determined and indeterminate beams. It calculates the reactions (both forces and moments), shear force and bending moment, as well as deflections for any of four typical beam end-fixing conditions (simply supported, cantilever, propped and built-in). The dimensions of the beam, the loads and their location are given by the user. It also makes it possible to choose the modulus of elasticity. The spreadsheet works as a menu driven program and is designed to be user-friendly [7].

COMPARISON BETWEEN MATLAB AND SPREADSHEET

A comparison between MATLAB and Excel is necessary to establish the contrasts in features, program construction, operation and quality of solution. MATLAB is numerical analysis software while Excel is a type of spreadsheet. In terms of program construction and expected results, MATLAB provides more convenient and adapted tools than Excel. Thus, MATLAB provides learners with a more effective computational tool to solve beam problems. However, the high cost of procuring MATLAB places it beyond the reach of most learners, leaving Excel as the obvious option. Excel is readily available since it belongs to the Microsoft Office pack.

In terms of difficulty in programming, Excel is more favourable than MATLAB. Excel spreadsheet calculation is easier and more intuitive than MATLAB. Excel provides an interactive interface and several easy to use tools and functions. The user does not need to go deep with program codes, although Visual Basic has a provision for that. Even without Visual Basic, Excel offers a convenient system called Macro, which displays all calculations and results in the same worksheet. It is not necessary to build a user interface like in MATLAB, and input data files and results displays can be incorporated into the worksheet. Thus, the independence of every step makes bug identification quite easy.

Contrary to MATLAB, handling of the software is easy and less time consuming. Working in MATLAB is more complicated, because it is exclusively numerical programming software. Methodically, it is possible to manipulate the program through M-files, which allow the splitting up of general codes into subroutines. Debugging functions provide a way to identify mistakes within the codes. Hence, MATLAB provides a more difficult way to establish the program and is more time consuming.

In terms of the method of programming, the same method was used for the two programs based on Macaulay and singularity functions in the case studied. It is only the operation that was different. Tools used to solve systems of equations for reactions at supports were the same. In MATLAB, a *for* loop is used to calculate these functions all along the beam and the smoothness of the curves to be plotted depends on how close the discrete points are made. Thus, it is important to be able to modify the number of divisions on a beam.

In Excel, without using programming, it is difficult for users to change the proximity of discrete points. Another option is for the programmer to make a definitive choice of good divisions compatible with all kinds of problems, precise enough to get smooth curves and not too many to avoid a long computation time.

In terms of speed of construction, the time dedicated for MATLAB was longer than that for Excel. Excel programs remain quick to construct and use due to incorporated functions and user-friendliness of the environment. However, use of Visual Basic within Excel provides a means to get best interface and more browsing buttons.

In terms of input data and directions, it is necessary to read figure tallying with beam type to correctly fill data files in both MATLAB and spreadsheet. By default, all data are equal to zero at the start of the program in MATLAB. On the contrary, Excel keeps the last processed problem in memory. Spreadsheet presentations can be modified through tools, tables, tabs, dialog boxes and changing colours. However, interactivity between the user and spreadsheet remains limited.

A difference of speed in input data can be also noted between Excel and MATLAB. In MATLAB, data entry is fast because of the independence between the files. On the contrary, Excel requires some few seconds between data entries for all spreadsheets with formulas involving data to refresh the new values.

Comparison of results from the two programs used in the case study revealed that both MATLAB and Excel gave the same results. Thus, learners can use any one of them without compromising quality and accuracy of output. Figure 5 is an output example of shear force diagrams for a built-in beam. The curves are similar, though finer in MATLAB.

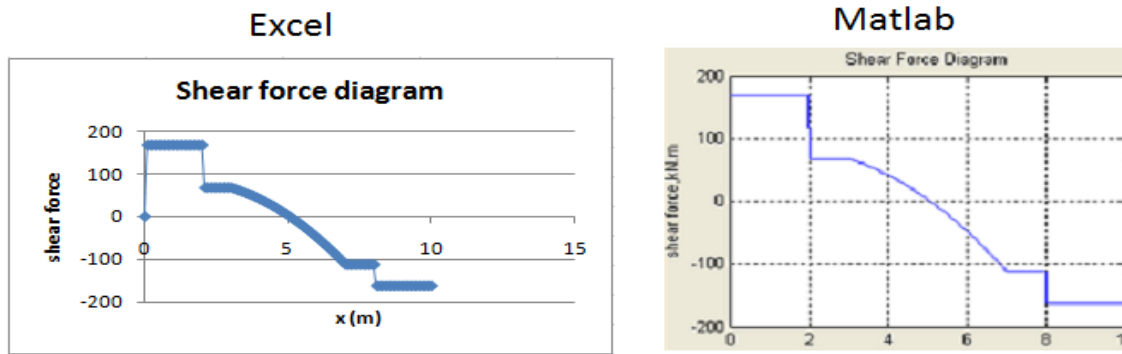


Figure 5: Shear force diagram.

CONCLUSIONS

In this article, the authors compare the features, operation and results of an Excel spreadsheet and MATLAB as computational tools to solve beam problems. Beam calculations are an important part of the engineering field, particularly, mechanical and structural engineering disciplines. Those calculations and beam theory constitute a large and fundamental part of any engineering curriculum. Both MATLAB and Excel can be used as an application for students to help them in the learning process.

The software based solution for a simple undergraduate design and analysis problem, in the form of a MATLAB script, can be easily used to analyse and solve a beam-bending problem subjected to different loading conditions. The actual MATLAB script enhances the educational benefits to students by giving them a tool to play with as they visualise the effects of changing loading conditions on bending moments, reactions, shear force, bending stress and deflection, as well as their locations. MATLAB provides a good programming platform through its graphical user interface that enables users to visualise the effects of changing loading conditions. Despite its advantages, MATLAB is more difficult to learn and to acquire than the spreadsheet for most learners.

The spreadsheet solution also provides a tool that might be used as a learning aid for students. MATLAB and Excel alike allow the students to introduce variations in the input data and, then, observe how changes in data affect the response.

However, the Excel solution gives a user-friendly instrument, which is not viewed by students as a *black-box* that they could not explore. The transparency of a spreadsheet tool helps to avoid any student perception that the work is something beyond his or her reach. It gives the student the ability not only to manipulate the problem for various conditions, but also to see how the calculations are done. Additionally, spreadsheets are easy to use and program construction is done without much difficulty. Excel's Visual Basic provides more useful tools that enable the creation of more comprehensive and easy to use programs.

Both MATLAB and spreadsheet solutions of this typical mechanics problem showcase the general importance of interactive, editable solutions to engineering problems. The MATLAB solution is faster and has a better user interface than the spreadsheet, however, its use by students is limited by its high cost of purchase. The advantage of the spreadsheet solution lies mainly in the general access of a typical Excel spreadsheet, which is easily available normally without any extra cost. However, its simplicity and ease in use provides an extra edge.

REFERENCES

1. Case, J., Chilver, L. and Ross, C.T.F., *Strength of Materials and Structures*. (4th Edn), London, UK: Arnold Publishers (1999).
2. Gere, J.M. and Timoshenko, S.P., *Mechanics of Materials*. (4th Edn), Boston: PWS Publishing Company (1997).
3. Jong, I.C., An alternative approach to find beam reactions and deflections: method of model formulas. *Inter. J. of Engng. Educ.*, 25, 1, 65-74 (2009).
4. Hanselman, D. and Littlefield, B., *Mastering MATLAB® 5*. The MATLAB® Curriculum Series, New Jersey: Prentice Hall (1998).
5. Nakamura, S., *Numerical Analysis and Graphic Visualization with MATLAB®*. New Jersey: Prentice Hall (2002).
6. Shine, S.J. and Sebusang, S.E.M., MATLAB as a teaching and learning tool for beam bending problems in mechanics. *Botswana J. of Technol.*, 13, 2, 12-19 (2004).
7. Uziak, J., Gandure, J. and Martin, D.L., Spreadsheet application in beam bending calculations. *World Trans. on Engng. and Technol. Educ.*, 12, 3, 402-407 (2014).