

## **Improving Machine Design Instruction by Developing Computational Design Tools**

### **Dr. David A Trevas, Northern Arizona University**

Lecturer in Mechanical Engineering at Northern Arizona University, 2016-present. Visiting Assistant Professor in Engineering at the University of the Incarnate Word (San Antonio, Texas), 2015-16. Held various positions in mechanical engineering and computer programming at Exxon Production Research, University of Texas Medical School, Halliburton, Baker-Hughes, GE Oil & Gas, and Cooper Power Systems.

### **Dr. John T. Tester, Northern Arizona University**

Dr. John Tester is a Professor in Mechanical Engineering at Northern Arizona University. His educational responsibilities are primarily in Engineering Design and Manufacturing. Dr. Tester has conducted funded research projects in biomechanics and engineering education. Dr. Tester's scholastic interests frequently integrate Undergraduate Engineering Education topics, typically in the area of the design of interdisciplinary engineering courses and curricula.

# Improving Machine Design Instruction by Developing Computational Design Tools

## Abstract

In a traditional machine design course, the analysis of machine components and systems involve lengthy, multi-step processes that often include retrieving data from graphs or tables. Automating these processes is quite feasible with computational tools that are ubiquitous in the marketplace and is routinely part of engineering work in industry. The ability to synthesize a solution algorithm from the reference equations and data obtained from tables and graphs shows that students understand the processes used in machine design. Once students master the fundamental machine design theory, they can create the computational tools and then proceed to engage the topics in machine design at a much higher level than with only use of pencil, paper and calculator. They can quickly vary input parameters into their computational tools and immediately see the influence on the output values of interest. In other words, they can actually design machine components and systems, instead of merely analyzing them given the input quantities for equations.

This work extends these previous findings of those using Microsoft® Excel in Engineering classrooms by teaching students how to build successful machine design-dedicated tools. Teaching students how to build these tools involves 4 basic steps: 1) introducing the machine design topic traditionally, 2) developing techniques for data input, 3) Algorithm development of the theory-based equations, and 4) validation processes required to insure future output accuracy.

The primary result of this approach is that students successfully answer questions on tests which are at a much higher level than those that were presented on previous tests. Students who have taken the course when they learned the techniques of developing computational tools have reported that they were able to put these skills to use at work or in their subsequent courses. This approach has the benefit of teaching marketable skills that also enhance the students' understanding of the concepts of machine design.

## Introduction

The machine design course in question is the first semester of a two-semester sequence in the topic area. The first semester is a junior-level course required for the mechanical engineering degree, while the second semester is a subsequent elective taken by either juniors or seniors. The textbook supporting both of the machine design courses is the popular and perhaps “the standard”, Shigley's Mechanical Engineering Design, 10<sup>th</sup> Ed.<sup>1</sup> This course has as its pre-requisites: statics, dynamics, mechanics of materials and materials science. The first-semester course covers threaded fasteners, shafts, springs, spur gears, and the mechanics of materials concepts that apply to these components.

As noted earlier, the calculation processes are very lengthy and error-prone. In a classroom session time, very few problems can be worked out. Many students emerge from the course with

only rudimentary skills in analyzing machine components without moving on to designing machine components for applications. To be able to design machine components, one must be able to vary parameters (dimensions, materials, heat treatments, etc.) and observe the effects that those variations have on output quantities such as factors of safety. Automating the analysis process will allow students to get experience designing.

Using computational tools to automate these complicated, time-consuming analysis processes is mentioned in the textbook, but this suggestion takes for granted that the students have these skills. Our experience has shown that few students entering this course possess the programming skills required for conducting this work in a timely manner, yet such skills are very useful in real-world engineering practice. We determined that integrating machine design with programming the students' growth will be enhanced by our presented approach in both machine design and automating needed analyses for that activity.

Microsoft ® (MS) Excel is the best tool for for many of the analysis processes introduced in this course (the justification for this claim will be presented later in this paper) but students need to gain competency beyond what may be considered as Excel basic knowledge.

## Background

The use of spreadsheet software as an education support tool in the machine design topic is not new. In 1988, Banta documented the use of Lotus 1-2-3 as applied to Mohr's circle and fatigue Failure.<sup>2</sup> The author noted three main advantages, being the need for progressing through many iterations of a fatigue failure analysis, increase reliability of calculation when compared to "hand calculations," and taking advantage of spreadsheet graphical output. The use of spreadsheets continued into the present century; Lin and Crossman used Microsoft Excel to accomplish similar machine design activities.<sup>3</sup> The ability to conduct multiple design iterations was again noted, with the performance criteria being fatigue analysis. They noted the improvement in reliability of calculations using spreadsheets rather than with the use of electronic calculators. They also added an additional observation: The students appeared to be able to better organize problems via the use of spreadsheets. However, they also cautioned of students simply running a program without deeply considering the theory behind the program. Tollner and Huang used spreadsheet software to teach bearing and shaft design.<sup>4</sup> Providing detail into the equations used, they noted the students using the software for a month as they completed their projects in that topic. Penrod also noted the need for reliable calculations with regards to spur gear train and shafts design.<sup>5</sup> He cited 27 specific design criteria the students needed to address, clearly illustrating the need for repeatable and reliable calculation of performance for different student designs. Penrod concluded with the acknowledgement of developing such comprehensive spreadsheets being initially time-consuming, yet insisted it was necessary due to the inherent iterative process of gear design. Ansaf and Jaksic recently provided details on how the use of spreadsheets in their machine design course allowed for a systematic instructional approach within their Mechatronics degree program.<sup>6</sup> Students addressed some weaknesses in basic design concepts and methodology by being required to develop the spreadsheets "from scratch" for several projects that addressed machine design topics.

This overview of spreadsheet use in a Mechanical Engineering Machine Design class illustrates the value of such an application as a means by insuring repeated calculations are reduced in error over design iterations. The primary concern discovered within this overview is the fear that students will lean too heavily on spreadsheets, once created, as a way to avoid learning deeply the theory. A similar concern may also be that students will not be able to apply the theory to new applications if they lean too much on the computational tools.

Here are some of these beliefs and the counter-arguments:

1. *Excel is for accountants and other financial types.* It is true that the spreadsheet concept originated from the world of finance, but it has been widely adopted for engineering design tools. Many companies distributed Excel spreadsheets as design tools to help engineering apply their products correctly. In our class, we demonstrate the worksheet provided by Key Bellevilles<sup>7</sup> that shows how to configure and select Belleville springs (also called washers.)
2. *Excel is not powerful enough for engineering.* People who say this are often not aware of the parts of Excel that are hidden by default. Realizing that most ordinary users will never write any Visual Basic for Applications (VBA) code, so the access to the Developer tools, including the VBA Editor are normally disabled. This can be enabled with a click on an Option/Preference. Other powerful facilities are distributed as Add-Ins which are included with Excel, yet users must turn them on to have access. Solver is the most notable add-in. While Excel is not appropriate for finite-element analysis or large linear algebra problems, it can be applied successfully to many kinds of engineering analysis processes.
3. *Excel is only for Windows computers.* This is not true, in fact, Excel was originally created by Microsoft for Apple Macintosh computers before Windows existed. At the time, Microsoft's operating system was DOS, and Lotus 1-2-3 was the dominant spreadsheet program on that system. The truth is that Excel works well on Macs today. There are a few features that are exclusive to the Windows version, such as ActiveX controls and some add-ins.

Using MS Excel in particular for education has three specific advantages:

1. *Ubiquity in industry.* Most mechanical engineering students are preparing to work in industry. When they start their career, it is almost certain that they will have access to Excel immediately. Many workplaces provide access to specialized tools, but that varies from one workplace to another. An engineer must justify the expense for specialized software beyond what may be currently available from the company. Using MS Excel as part of an MS Office suite is a quick approach to starting a new engineering project.
2. *Extensive support resources.* Since Excel is used for a wide variety of applications beyond engineering, there are many resources to learn how to use Excel and solve problems. There are far more books, websites, videos, etc. on Excel than there are for any engineering application software. There are many explanations of the functionality of Excel from different perspectives, which also may lead to better understanding.

3. *VBA (Visual Basic for Applications) and Add-Ins.* Many sophisticated tools can be developed when the power of add-ins and functions created in VBA code. Additionally, engineers have been able to create entire data-acquisition (DAQ) systems in Excel.

Excel does have some limitations that should be considered when choosing which application to use for a task.

1. *Performance can lag with large problems.* Excel runs more slowly than compiled code, but this is not perceptible until the spreadsheets get very large and the sequences of calculations get long. The limits are moved as computer capability expands, and the tools programmed in machine design class never tested the limits of practical application.
2. *Tools are easy to damage accidentally.* One feature that makes Excel easy to use also allows for a macro or software tool failure. Cells on the spreadsheet are eligible as locations for data entry or calculation. The usefulness of Excel is in the cells that make calculations – these locations are where the theory and equations are implemented. If a user overwrites a calculation cell with a data value, and then saves the worksheet, the intelligence of the cell is permanently lost. Learning to unlock data entry cells while protecting the rest of the worksheet is a best practice that should be used to prevent this problem from happening.
3. *Some terms are not presented in the language engineers are familiar with.* With its heritage from finance and later application to engineering, some of the terminology conceals functionality from engineers simply by having unfamiliar names. The solution is that users must learn the terminology and adapt to the unfamiliar names. Examples of these are given in Table 1.

Engineering	Excel
Curve fit	Trendline
Linear regression	LINEST()
Linear interpolation	FORECAST.LINEAR()
Mean	AVERAGE()
Integer	Whole number
Floating-point number	Decimal

**Table 1. Engineering terms in Excel terminology**

Considering the above context, we decided that MS Excel would be the most suitable application to use for teaching those students who are preparing for entry-level engineering jobs in industry how to automate design tools. The motivation of this course is not to merely give students the four design tools for the machine components we cover in this course, but to also provide them the ability to build a software design tool from scratch for problems they will encounter in the future. General courses in Excel cover nearly all of the capabilities of the program, some of which include many features that are not particularly useful for this purpose; other courses may neglect any features that would appeal to engineers. In the following section, we present the concepts we stressed to guide students in learning how to automate analysis processes in machine design.

Process

A primary principle of using computational tools to teach machine design is that teaching the fundamentals of the subject traditionally and having students go through the problem-solving process manually are necessary before moving on to automating those processes. Handing the students pre-written worksheets is not acceptable. The major skill we seek to develop with this method is the ability to appreciate the problem-solving process and being able to conceive algorithms to automate the process that yields answers quickly and accurately.

While instructing students about the phenomena that appear in machine design, pointing out what steps are being used will inform them about how to automate the tasks in the future. Each equation used can be considered as the basis of a spreadsheet formula. Quantities such as material properties are often extracted from tabular data, and this suggests that a lookup table could be implemented. There are factors that are extracted from graphs. For example, stress concentration factors are shown in graphical form<sup>8</sup>. One way to make working with these quantities efficient and accurate is to digitize the graphs and use curve-fitting tools to create formulas to make calculations. Occasionally, there are relationships among parameters that are too complex to be conveniently calculated in a single spreadsheet cell. These can often be handled by using the underlying programming environment, VBA (Visual Basic for Applications), that is hidden by default to Excel users.

Once students have completed some analysis procedures by hand, they should have some insight of what the key input and output quantities are. It is important to make reasonable choices about which quantities to consider as input and which to consider output, so the tool is convenient to use in most cases. Anyone who has been involved with problem-solving knows that sometimes quantities can be either input or output. In the case where the selection of input and output is backwards, Excel has Goal Seek and Solver capabilities to solve those cases.

The general process is to start with the input quantities in a set of columns in the upper-left part of the main worksheet. Below the input quantities, continue with intermediate values that are required to make the final calculations. Then, starting at the top of the main worksheet and to the right of the inputs, make the next set of columns the output values. Once the tool is completed and validated, the input and output values can be seen on a single screen, so the typical user experience is convenient and easy.

The design of the tool starts with the desired policy about units. There are three variations:

1. Use one set of units (metric, SI, inch-pound-second, etc.) and make it the user's responsibility to do the conversions on the input and output values.
2. Allow the user to select a system of measurement and do the calculations in the various systems.
3. Allow the user to select a system but convert all input units to the base units for calculation, and then convert the base units to the preferred units for the outputs (and optionally for the intermediate values.)

For the first two options, the number of columns selected should be five, but for the third option, the number should be seven. The purpose of each column is:

Column 1: The name of the quantity

- Column 2: The symbol for the quantity, if needed
- Column 3: The value of the quantity
- Column 4: The unit associated with the value, if needed
- Column 5: A reference. This could be an equation, table or figure.

An example of this format is illustrated in Figure 1.

For the 7-column format, the remaining columns are:

- Column 6: The value converted to the base units
- Column 7: The base unit associated with the base value.

Name of quantity	Symbol for quantity	Value of quantity	Unit associated with quantity	Reference
Units		Metric		
Material		Music wire		Tables 10-4, 5
Ends		Squared and ground		Table 10-1
Peened?		FALSE		Eq. 10-28, 29
Wire diameter	d	1.4 mm		
Outer diameter	OD	11 mm		
Coil diameter	D	9.6 mm		Figure 10-3
Spring stiffness	k	9.5999 N/mm		Eq. 10-9
Fractional overrun	$\xi$	0.5		Eq. 10-17
Maximum force	Fmax	80 N		Eq. 10-17
Minimum force	Fmin	60 N		Eq. 10-31
Active coils	Na	4.65		Eq. 10-9
Total coils	Nt	6.65		Table 10-1
Solid length	Ls	9.307 mm		Table 10-1
Force to solid	Fs	69 N		Eq. 10-17

Figure 1. Screenshot of Helical Compression Spring Design Tool. (Five-column format)

The reference column (#5) is used to insure quality of the design tool. It is good practice to refer to the source of an equation or of data obtained from a table or a graph. Often, equations are accompanied by text describing the limits of applicability, and those who may eventually use the tool should be able to consult the source material to understand they are using it appropriately.

To make a tool robust, Excel has data validation tools to filter values that can be placed in cells so that only valid values are entered. For example, the numbers of gear teeth can be forced to be integers (whole numbers, in Excel parlance) and between minimum and maximum values. Values that come from lookup tables can be guaranteed to be valid if the cell is restricted to the list of entries in the data source.

One problem that plagues this approach is that, in the default state, it is easy to overwrite a cell that is calculated with a formula with a hard-coded value. At this point, if the spreadsheet is saved, the functionality of that cell is lost. The solution to this is straightforward but is cumbersome during the development process. By default, all cells on an Excel spreadsheet are locked, but sheets are unprotected, so there is no effect. To make this feature helpful, the student should highlight the cells that are intended to be input values and unlock them. Once that is completed, engage the Protect Sheet function and this prevents any cells other than the unlocked inputs from being altered. To prevent yourself from absent-mindedly overwriting a calculation

cell, it is easiest to use no password when protecting the sheet. A warning will pop up, but you can easily Unprotect the sheet if that is what is desired. Protecting a sheet with a password makes it more secure, but there is no recovery if the password is lost.

Once the students are familiar with the problem-solving process and have decided on a strategy designating which quantities are input, output and intermediate, the calculations need to be programmed into the value cells. There are many resources to learn how to enter basic formulas, so this will not be discussed further here.

Engineering formulas are often piecewise functions, that is, different equations apply for certain ranges of the input variable. The Excel IF() function is useful to handle those cases, and if there are more than two segments of the function, nested IF() formulas are necessary.

There will be cases where a function requires more calculation than what can be practically performed in a cell formula. For the past few years, Microsoft has decided that most ordinary users will never get to this level of difficulty, so the Developer tab is disabled by default. Once this tab is enabled, the student has access to the Visual Basic for Applications (VBA) Editor. Here, one has access to a full-fledged programming language where very complicated functions can be created. Unfortunately, the syntax of VBA is not the same as languages that are related to C, so one will need reference materials<sup>9</sup> to get started with the language.

Before adding VBA functions to your design tool, there are some issues to consider that make using them a little annoying to use. Historically, VBA macros have been used in malware attacks, so Excel has added security measures to combat them. First, to save a design tool with VBA code, it must be saved with a .xlsm extension as opposed to the standard .xlsx extension. Then, every time one opens this file, a dire-sounding security warning pops up and the user has to allow the design tool to access the VBA functions.

On the other hand, including VBA code has great advantages. If one wants to keep their methods confidential, or are dealing with some proprietary data or algorithms, the code modules can be password-protected to prevent others from seeing their contents. Also, the functions cannot be damaged by overwriting, whereas unprotected cell formulas potentially can be damaged. While performance can be slowed in extreme cases, it is possible to write extensive programs without having a noticeable impact on the worksheet responsiveness.

To complete certain calculations in Machine Design, quantities must be retrieved from tables or graphs. This would appear to prevent a complete calculation process from being fully automated. However, there are approaches to overcome these obstacles. For graphical data, there are tools available to digitize graphs that allows students to convert curves on a graph to a series of digital points. One popular tool for this is WebPlotDigitizer<sup>10</sup>. It allows one to enter a graph in a common image format such as JPEG, PNG, GIF or BMP, and extract data in a comma-separated variable (CSV) format which can be opened up directly in Excel. From the extracted data, a chart can be generated in Excel and from there, the trendline facility can be employed to examine curve-fits of the data.



With the trendline function, curves of various types (linear, logarithmic, exponential, polynomial, power law) can be examined and the goodness of fit can be judged visually and by considering the  $R^2$  value associated with the least-squares fit. Once the best fit is found, the equation can be displayed and used as a formula in the design tool.

Tabular data can be easily handled in an automated design tool, but the approaches vary based on the nature of the data. Types of tables encountered include:

1. A series of independent records. An example of this form is a table of physical properties. The left-hand field of each record is the name of a material and subsequent fields contain the Young's Modulus, the Poisson's ratio and the weight per unit volume of the material. To retrieve the properties, use the VLOOKUP() function where the material name is the search criterion. The list of material names should be available as a drop-down list by using Data Validation.
2. A series of stepwise functions. An example of this is the table of International Tolerance Grades ([ANSI B4.2-1978](#)) for limits and fits. For a range of diameters, each grade of tolerance is defined as a fixed value (e.g., for diameters ranging from 30-50 mm, the International Tolerance grade 7 is 0.025 mm.) To retrieve this data, use the MATCH() and INDEX() functions. The input value should be restricted to be a floating-point number (a Decimal number, in Excel parlance) within the lower and upper limits of the table.
3. A two-variable function. The horsepower ratings of standard V-belts as a function of sheave diameter and belt speed is an example of such a function found in Machine Design<sup>11</sup>. The left-hand column of the table contains several sheave diameters and the top row consists of selected values of belt speed. In general, the sheave diameter will not coincide with one of the listed values, nor will the belt speed exactly match the few values listed. Using MATCH() and INDEX(), the values of horsepower that bracket the diameter and speed selected can be found and linear interpolation can then be used to make a good estimated of the horsepower corresponding to the input quantities. Excel has a built-in linear interpolation function called FORECAST.LINEAR() that can make those calculations easier.

One way to verify that a computer code is correct is by finding a correctly-solved problem and entering the input values and seeing whether the outputs match. The problem with this approach in machine design is that many of the processes we automate will have many intermediate steps between the inputs and the final results. To make the debugging process easier, the students are given a number of fully-solved problems including all of the calculations, including the intermediate ones. This way, the student can detect where the process goes awry and can fix the problems until the whole process is validated. Depending on whether one or two systems of units are being used, the number of complete examples provided in a validation file is typically in the 6-10 range.

In the ideal case, all of the problems that students will attempt to solve with their design tools have exactly the inputs in the problem statement that appear in the input section of the worksheet. The reality is that there can be some deviation and ways to apply the design tool are presented here.

One common example of a deviation that can be encountered is that the problem states a target value for an output value and asks to find the input value that would result in the target. A case of this could be a compression spring design tool where the factor of safety is an output value and the wire diameter is one of several inputs, and the problem asks for a wire diameter that results in a specific factor of safety. The Solver add-in can be used to find the answer very efficiently. When tasked to find a single value to achieve a target, Solver works very reliably. It can also be used to solve multi-variable problems, but sometimes it fails to converge on a solution.

Another issue that arises is that problem statement gives a value that is a calculated intermediate in the design tool. If one uses the original design tool and overwrites that formula (after unprotecting the sheet), that formula becomes permanently lost. The way to get around this is to use Save As... to create a copy of the original spreadsheet. With this copy, overwriting a formula with a hard-coded value is needed for this problem alone, so it is appropriate to make this separate copy to address the problem at hand. This came up in an AGMA method gear design problem<sup>12</sup>. The main design tool contained a method to calculate the geometry factor. In one problem statement, the geometry factor was given, so a separate copy was created where the method was overwritten by the values given in the problem statement.

## Results

To test the efficacy of this approach, using computational design tools in a first-semester machine design course, the course grades before and after the approach was implemented were compared. With the tools developed in the latter class, the questions posed on exams to those students would not have been possible for the first class to answer in a reasonable time.

For example, to calculate factors of safety for spur gears designed with the AGMA method, about a dozen intermediate factors need to be found. On the test given to students using just pencil, paper and calculator, a test question gave them all but three of the intermediate factors and asked them to calculate those three and then use the results to determine the gears factors of safety. The latter class was challenged to find the face width to achieve a specific value for factor of safety, with no intermediate factors given directly.

One metric used at our university is the ratio of students that withdraw, fail or earn a grade of D, to the number of students that enrolled at the beginning of the semester, and this is called the DFW rate. Before the computational tools were brought into the machine design class, the DFW rate was 17%, and after the current approach was implemented, the DFW rate fell to 12%

In course evaluations, most students are happy that they have learned Excel skills to address machine design problems, but a few complain. Without being able to associate course

evaluations with the students' performance in the course, it is difficult to know if the complaints are valid or if they are just the venting of disgruntled students who did poorly in the class.

There is plenty of anecdotal evidence that students are applying what they have learned in subsequent classwork. For example, several of the teams in the senior capstone design course have been using Excel design tools in their work. Also, several students who have gone into summer internships or even entering the workforce have reported applying the Excel skills learned in this class to their work.

## Conclusions

We conclude that the approach presented here -- teaching students in a machine design course how to build and use design tools in Excel – has several benefits to contribute to their success. Being able to relate to the material at a higher level, that is, spending more time on thinking about the design of machine components and less time on long tedious analysis calculations, they are able to contribute more to design tasks at their future work. Competency at building reliable design tools will also enhance their value to people with whom they work. Being able to distribute robust, accurate design tools to one's colleagues or co-workers makes the student a more valuable employee.

## Bibliography

---

<sup>1</sup> Budynas, Richard G; Nisbett, J. Keith, *Shigley's Mechanical Engineering Design*, Tenth Edition, McGraw-Hill, 2015

<sup>2</sup> Banta, Andrew, "Using Computer Spreadsheets in Engineering and Education," *Frontiers in Education Conference Proceedings*, 1988.

<sup>3</sup> Lin, C.Y., Crossman, G. R., "Using Spreadsheet Programs in Teaching Machine Design", *Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition*,

<sup>4</sup> Tollner, Ernest W.; Huang, H. Y., "A shaft design aid for integrating basic elements of introductory machine design," *ASEE Annual Conference Proceedings*, p 12835-12845, 2004, *ASEE 2004 Annual Conference and Exposition, "Engineering Education Research: New Heights"*

<sup>5</sup> Penrod, James, "An innovative approach to a classic design project," *ASEE Annual Conference and Exposition, Conference Proceedings*, 2006

<sup>6</sup> Ansaf, Bahaa Kazem ; Jaksic, Nebojsa I, "Teaching mechanical design for mechatronics engineering students using a project-based sequential learning approach," *ASEE Annual Conference and Exposition, Conference Proceedings*, v 2018-June, June 23, 2018

<sup>7</sup> Key Bellevilles, Inc., "Design Program" [Microsoft Excel spreadsheet], Leechburg, PA, 2018.

<sup>8</sup> Pilkey, Walter D.; Pilkey, Deborah F., *Peterson's Stress Concentration Factors*, 3<sup>rd</sup> Edition, Wiley, 2008.

<sup>9</sup> Urtis, Tom, *Excel VBA, 24-Hour Trainer*, Wiley Publishing, 2011.

<sup>10</sup> Rohatgi, Ankit, "WebPlotDigitizer", <https://automeris.io/WebPlotDigitizer/>

<sup>11</sup> Budynas, Richard G; Nisbett, J. Keith, *Shigley's Mechanical Engineering Design*, Tenth Edition, McGraw-Hill, 2015, Table 17-12.

<sup>12</sup> *Fundamental Rating Factors and Calculation Methods for Involute Spur and Helical Gear Teeth*, ANSI/AGMA 2001-D04, 2001.