

# Creating a Contextualized Learning Environment in a Computational Methods Course

David C. Che

*Bryan College*

## Abstract

Traditionally, a Computational Methods course focuses on skills development through the teaching of a programming language and/or the use of Excel spreadsheet. Sometimes numerical analysis topics are briefly covered as well<sup>1,2,3</sup>. An attempt was made at Bryan College in the fall of 2019 to provide a contextualized learning environment in the sophomore level computational methods course that would expand students' exposure to some highly practical and useful fields of systems engineering, operations research, design optimization, management decision making, and manufacturing quality control. The concept of "mini-courses within a course" is discussed. Used properly, it could serve as another way to enrich students' learning experience while adding width and depth to the curriculum. Learning outcome assessment results are shared and challenges in using this approach discussed. Student feedback was positive.

## Key Words

Linear programming, computational methods, contextualized learning, Excel spreadsheet, Matlab programming.

## Introduction and Motivation

Computational Methods (EGR-221) is a sophomore level required core course for all engineering majors at Bryan College. This is the current catalog course description:

*Students are introduced to some of the key computational techniques used in modeling and simulation of real-world phenomena, including error analysis, matrices and linear systems, convergence, and stability.*

This course was first offered at Bryan College in Fall of 2019 as part of the new BSE program. Prerequisites for this course are Calculus I (MATH-122) and Introduction to Programming I (CSCI-116). In CSCI-116, students are taught C++ (a 3 credit hour course taught by computer science faculty). It was the intent of the engineering faculty that EGR-221 Computational Methods focus on teaching students a more practical software/tool such as *MATLAB* and/or Excel.

Within the BSE program students have options (concentrations) in mechanical engineering, civil engineering, and computer engineering. An attempt was made to strike a balance between teaching the use of modern engineering tools such as *Microsoft Excel/MATLAB*, and the application of advanced math to problem solving in diverse fields of engineering to meet the needs of all students. The first half of the semester was devoted to the teaching of Excel, and the second half to the teaching of *MATLAB*. For Excel, some important subjects that are normally not part of a typical undergraduate general engineering curriculum, such as linear programming and

manufacturing quality control, were incorporated in the curriculum. Linear programming is useful in design optimization and business decision making, and manufacturing quality control is typically a part of an industrial/manufacturing engineering curriculum. With six sigma quality process certification and robust product design becoming more and more mainstream in any industry, these topics are useful for all engineers to learn. For *MATLAB* programming, an attempt was made to incorporate solving Statics problems extensively with *MATLAB* commands, especially when dealing with vector notations and operations involving dot product, cross product, and triple mix products.

### Contextualized Learning of Excel

When teaching Excel, the instructor devoted 3-4 lectures to covering some basics of statistical process control (SPC), sometimes referred to as statistical quality control (SQC). Traditionally, these topics are covered in an industrial/manufacturing engineering or engineering management curriculum, and with the advent of six sigma quality control and robust design philosophy in almost all industry and service sectors, it is becoming more important for all engineers to have some basic knowledge in these subjects. The SPC/SQC are used as a context to teach students how to use Microsoft Excel to input data and generate graphs and interpret them. Fundamentals are taught in creating charts such as histograms, a variety of bar charts, x-bar and R charts, scatter plots, and how to edit charts (adding titles, axis labels, formatting, etc.) Figure 1 shows a typical run chart generated from process data to the left. To generate this chart, with upper and lower control or spec limit lines and a center line, students need to understand some basic statistical concepts such mean and standard deviation. Students are not only learning a new subject matter, but are also using the Excel tool to implement the concepts learned. Course materials are drawn from popular SPC textbooks such as Montgomery<sup>4</sup> and DeVor, et al<sup>5</sup>.

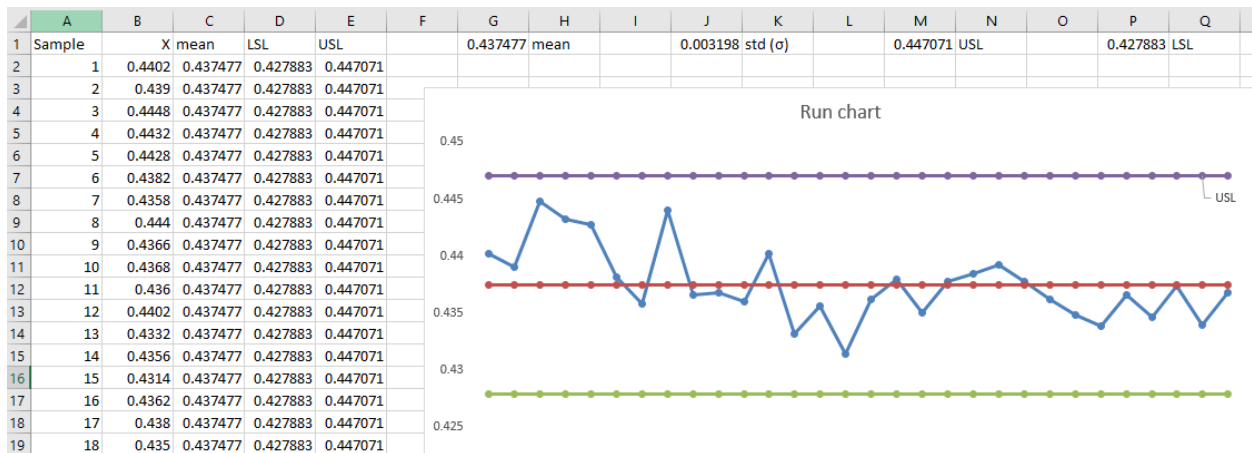


Figure 1 Using Excel Charting Tools to Generate a SPC Control Chart

Data analysis in Excel are also taught, with topics in linear regression analysis (adding trend lines, interpolations and extrapolation), interpreting quality of fit and display equations on charts, etc. Basic concepts of convergence, numerical techniques, etc. are covered when teaching Excel Data Solver. The use of Excel Data Solver has been emphasized in solving linear and nonlinear

equations or systems of equations. Examples from engineering mechanics - statics, are used to demonstrate the power of Excel Data Solver<sup>6</sup>.

In Fall of 2019, the author spent 3-4 lectures on linear programming (LP) topics in EGR-221. The topics were related to production decision making and design optimization<sup>7, 8</sup>. Sometimes LP is under the umbrella of “operations research,” or “systems engineering”. Occasionally, LP is covered in some advanced algebra curriculum of high school math<sup>9, 10</sup>. For college level courses, the author had found LP covered in systems engineering and management textbooks<sup>11, 12, 13</sup>. But the high school textbooks listed above are more accessible to freshmen and sophomore engineering students. LP problems can be easily set up as systems of inequality equations, which lend itself to easy solution with Excel Data Solver. They are basically constrained optimization problems: finding maximum or minimum values under certain constraints such as cost, capacity limit of machines, etc.

### **Contextualized Learning of MATLAB**

Students were taught *MATLAB* in the second half of the semester. The same students were also taking Statics concurrently that semester. It makes it natural to use *MATLAB* to solve statics problems. *MATLAB* is particularly powerful in doing vector operations, which is abundant in statics. The following were some course objectives involving *MATLAB*:

1. *MATLAB fundamentals*
  - a. *Use of vectors to solve statics problems*
    - *Dot product (projection of one vector onto another vector)*
    - *Cross product (moment of a force around a point)*
    - *Triple mixed product (moment of a force around an axis)*
    - *Center of gravity/center of mass/centroid of composite shapes/bodies*
  - b. *Use of matrices to solve systems of linear equations*
  - c. *Basic plotting and use of math/trig functions in MATLAB*
2. *MATLAB programming*
  - a. *Be able to write simple programs (including the use of functions)*

Two examples are shown in the Appendix.

### **Outcome Assessment**

Test #1 was conducted one-on-one. Test #2 was conducted in a small group setting so students could get help if they needed it. It is time consuming to do this one-on-one testing, but it is more formative than traditional group testing, as students learn in the process. In other words, since the students are working on the problems in front of the instructor, if they get stuck, they can ask for instant assistance and feedback. Even though asking for help means their initial assessment grade will be lower, they can move on to eventually solving the problem.

The following are some student feedback:

2021 ASEE Southeastern Section Conference

- The real world examples used by the instructor was very helpful for my learning.
- I like the one-on-one test style. I like the instruction methods in this class – the instructor goes through each step with students on the big screen, which is very helpful for me.
- I find the subject matter to be very interesting. My desire to learn more about software grows with each class.

Table 1 Mapping of Course Learning Objectives with ABET Student Outcomes (SOs)

Course Learning Objectives (CLOs)	ABET SOs
1. Students will be able to use Excel spreadsheet to solve engineering problems.	1
2. Students will be able to use MATLAB to solve engineering problems.	1
3. The students will have an ability to communicate research results or findings in a way that is understandable to others	3
4. The students will gain an understanding of numerical analysis	1

Table 2 ABET Student Outcomes vs. the Course Learning Outcomes (CLOs)

New ABET Student Outcomes	Correlated Course Outcomes	Key Assignments	E	A	M	U	Average
#1. an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics	CLO 1, 2, 4	Questions (1) and (2) of Test 2	5	9	2	2	2.94
#3. an ability to communicate effectively with a range of audiences	CLO 3	Question 2c of Test 1 on creating charts in Excel	8	7	2	1	3.22

**Sample Rubric<sup>14</sup> (on a Scale of 1.0 to 4.0):**

**Excellent (4.0):** student fully understands the problem and is capable of solving it correctly with no help

**Adequate (3.0):** student is able to solve the problem with some help or student is able to solve a big percentage of the problem without help

**Minimal (2.0):** student is able to solve the problem with significant help or the student is able to solve part of the problem with some help

**Unsatisfactory (1.0):** Student is unable to solve the problem even with significant help

## Summary and Conclusions

The teaching of MATLAB programming and Excel spreadsheet has been organically integrated with teaching solid engineering subjects such as statistical process control (SPC), engineering mechanics – statics, systems engineering and design optimization (linear programming, or LP). The mini-series on SPC and LP are like “mini-courses within the main course”. It has shown some promise in creating a contextualized learning environment for students.

The challenge is that it requires the instructor to have some background knowledge in these mini areas, and not all instructors in engineering are well versed in SPC and/or LP. An argument can be made that, either the instructor need to gain some knowledge in those areas, or pick another one or two areas of expertise that he/she does have and use those as teaching contexts.

The author of this paper had some work experience in the SPC field, so it came naturally; but he was a novice in LP as much as anyone. He has always wanted to cover some of the LP topics in the undergraduate curriculum, but it never happened until his daughter started to ask him questions about solving LP problems in her online Algebra II Honors course. He then used the textbook chapter that his daughter used to teach LP to his students, and supplemented it with other materials from systems engineering and design optimization textbooks. Students seemed to enjoy these topics, as reflected in their course surveys and informal interactions with the instructor later on. Formal assessment results also pointed to satisfactory performances of students in achieving student learning outcomes.

The challenge of using statics as a context to teach MATLAB is that unless the students are also taking Statics concurrently or prior to the Computational Methods course, it might be hard to go too deep into the vector analysis part of the course material. It is rare to see Statics being listed as a prerequisite or co-requisite to Computational Methods. To overcome this, if indeed not all students had taken or are taking Statics when Computational Methods is taught, the same idea of a “mini course within a course” can be implemented.

In fall of 2020 the author was not assigned to teach Computational Methods and the new instructor covered more numerical analysis with MATLAB. In some ways it is a return to the “norm” of what this course is typically about, but it meant the loss of some of the benefits of a contextualized learning environment that have been demonstrated by this study.

The “mini-course within a course” concept may still be implemented in other courses. For example, the author plans to use some of the same modules to cover Excel, LP and SPC in his freshmen Service Engineering class in the Spring of 2021. In some ways these mini-course modules are quite transportable. Once the instructor grasps the content knowledge, he/she can easily use it in other parts of the curriculum to enrich the students’ learning experience.

Physics educators at Brigham Young University had tried splitting the three credit hour computational methods course into three one credit hour “labs”, one for sophomores, one for juniors, and one for seniors. “Students are introduced to symbolic methods using Maple when they are sophomores, and to numerical methods using MATLAB beginning in their junior year. This helps prepare the students for their upper division courses, prepares them for the research they will do for their senior project, and spreads computational methods throughout the undergraduate curriculum<sup>15</sup>.” This might be another way to lessen the impact of the above mentioned challenges. As one reviewer commented, “the BYU approach is certainly one that has

merit - it is a scaffolded approach to learning computation methods. Definitely wish computational methods were woven more seamlessly into engineering curricula rather than smacking students in the face with it once and hoping they'll retain the knowledge by senior year.”

## References

1. Larsen, K. F., & Hossain, N. A., & Weiser, M. W. (2016), *Teaching an Undergraduate Introductory MATLAB Course: Successful Implementation for Student Learning*, 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana
2. Daniels, S., & Aliane, B., & Nocito-Gobel, J., & Collura, M. (2004), *Learning The Methods of Engineering Analysis Using Case Studies, Excel, And VBA*, 2004 Annual ASEE Conference, Salt Lake City, Utah
3. Dai, Q., & You, Z. (2006), *Using Matlab to Solve Engineering Problems for Undergraduates*, 2006 Annual ASEE Conference & Exposition, Chicago, Illinois
4. Montgomery, D. C., *Statistical Quality Control*, 4<sup>th</sup> edition, John Wiley & Sons
5. DeVor, R. E., & Chang, T., & Sutherland, J. W., *Statistical Quality Design and Control: Contemporary Concepts and Methods*, 2<sup>nd</sup> edition (2007), Publisher: Pearson/Prentice Hall
6. David C. Che, *Incorporating Excel Data Solver in Teaching Undergraduate Engineering Curriculum*, Proceedings of the 2016 ASEE North-Central Region Annual Conference, Mt. Pleasant, MI
7. J. S. Arora, *Introduction to Optimum Design*, (2017), 4<sup>th</sup> edition, Elsevier
8. P. Venkataraman, (2009), *Applied Optimization with MATLAB Programming*, 2<sup>nd</sup> edition, John Wiley & Sons, Inc.
9. James, F. et al, *Advanced Algebra*, the University of Chicago School Mathematics Project (UCSMP), 3<sup>rd</sup> edition, (2010), Wright Group/McGraw-Hill
10. Elayn Martin-Gay, *Algebra 2*, (2015), Pearson Publishing Company
11. M. Q. Anderson, & R. J. Lievano, (1986), *Quantitative Management – An Introduction*, 2<sup>nd</sup> edition, Kent Publishing Company, pages 41-52
12. Richard de Neufville, *Applied Systems Analysis – Engineering Planning and Technology Management*, (1990), *McGraw-Hill Publishing Company*, page 71
13. B. S. Blanchard, & W. J. Fabrycky, (2006), *Systems Engineering and Analysis*, 4<sup>th</sup> edition, Pearson/Prentice Hall, pages 276-278.
14. John K. Estell, et al, (2012), *Improving Upon Best Practices: FCAR 2.0*, ASEE 2012 Annual Conference, <https://peer.asee.org/improving-upon-best-practices-fcar-2-0.pdf>
15. Ross L. Spencer, (2005), *Teaching Computational Physics as a Laboratory Sequence*, American Journal of Physics, Vol. 73, Issue 2.
16. R. C. Hibbeler, (2011), *Statics and Mechanics of Materials*, 3<sup>rd</sup> edition, Pearson Prentice Hall
17. Sridhar S. Condoor, (2009), *Engineering Statics*, 3<sup>rd</sup> edition, SDC Publications

## Author Biography

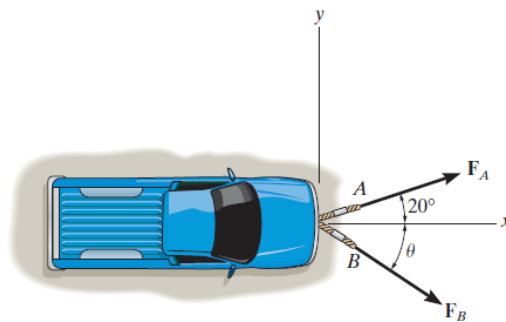
Dr. David C. Che worked in the industry for over ten years, including eight years with General Motors in Michigan, before beginning his full time teaching career. He first taught at Geneva College in western Pennsylvania for seven years, then at Anderson University in central Indiana for one year, and Mount Vernon Nazarene University in central Ohio for three years. He is now Chair and Professor of Engineering at Bryan College in eastern Tennessee. His research interests include CAD/CAM/CAE,

automotive engineering, manufacturing engineering, quality control, mechanical design, engineering mechanics and engineering education. He is a member of ASEE and ASME.

## Appendix Sample MATLAB Programming Problems Used in this Course

### Example 1

2-19. The truck is to be towed using two ropes. If the resultant force is to be 950 N, directed along the positive  $x$  axis, determine the magnitudes of forces  $F_A$  and  $F_B$  acting on each rope and the angle  $\theta$  of  $F_B$  so that the magnitude of  $F_B$  is a *minimum*.  $F_A$  acts at  $20^\circ$  from the  $x$  axis as shown.



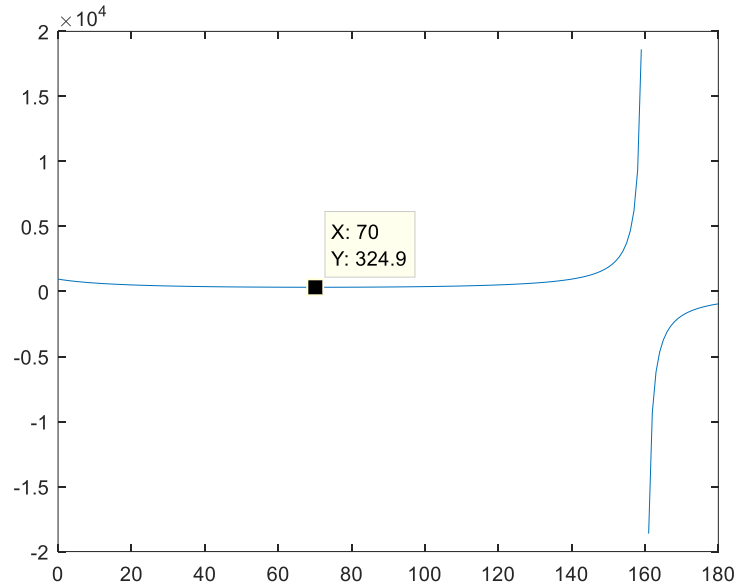
This problem was from R. C. Hibbeler's *Statics and Mechanics of Materials*<sup>16</sup>. Solution given by the solutions manual was graphical solution. The following presents a numerical solution.

**Solution:** Write out two force equilibrium equations and derive an expression of  $F_B$  as a function of  $\theta$  only. Plot it using Excel or MATLAB and find the minimum value of  $F_B$  and corresponding  $\theta$ .

Excel Data Solver can also be used to solve this. Here is a MATLAB solution at the command line:

```
>> theta=[0:180];
>> FB=950*tand(20)/(sind(theta)+tand(20)*cosd(theta));
>> FB=950*tand(20)./(sind(theta)+tand(20)*cosd(theta));
>> plot(theta,FB)
>> min(FB)
ans =
-1.8617e+04
>> min(abs(FB))
ans =
324.9191
```

Using the MATLAB built-in *Data Cursor* tool, we can find on the plot that  $\theta$  is 70 degrees when the minimum value of  $F_B$  is achieved.



Example 2

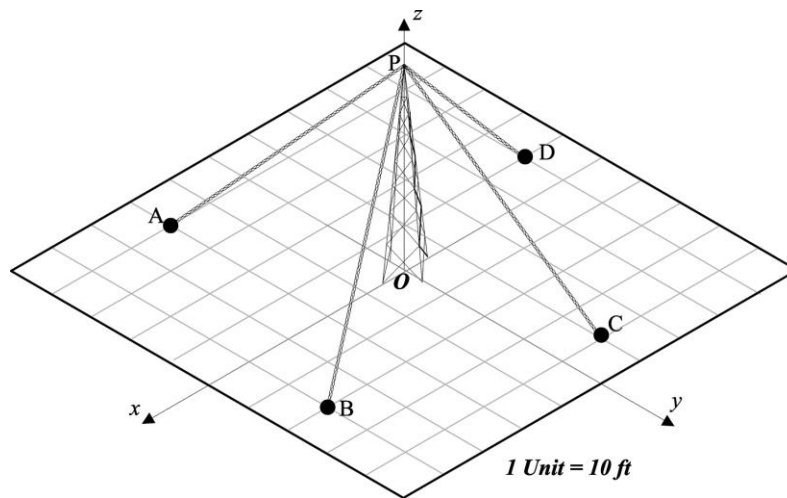


Figure 2 Sample Statics Problem<sup>17</sup>

*Rewrite the homework problem above into a MATLAB function with point coordinates (A through D), pole height (to get point P) and forces in each cable as input variables. Output variable should be the bending moment around the base, point O, expressed as both a Cartesian vector (in i, j, k format) and the magnitude. Use results from your graded homework as a test case.*



*Challenge: expand your inputs to include as many points as needed, in other words, eliminate the four-point restriction, so that the function can take more than or less than four points as anchors on the ground.*

*A 40 feet tall tower is stabilized in a plane using four cables. The tension in each cable is 800 lb. Find the net bending moment at point O using MATLAB.*

MATLAB solution for this problem:

```

% position coordinates of each point:
A = [20 -40 0]; B = [40 20 0]; C = [-10 40 0]; D = [-40 -10 0]; P = [0 0 40];
% vectors for each cable:
PA = A-P; PB = B-P; PC = C-P; PD = D-P;
% unit vectors for each cable:
Upa = PA/norm(PA);
Upb = PB/norm(PB);
Upc = PC/norm(PC);
Upd = PD/norm(PD);
% force vectors for each cable:
Fa = Upa*800; Fb = Upb*800; Fc = Upc*800; Fd = Upd*800;
% sum of all forces:
F = Fa + Fb + Fc + Fd;
R = P;
M = cross(R, F) % cross product to find bending moment M = R x F

```

A more generic solution:

Input **X** is a stacked matrix of anchor position coordinates, it needs to be in the format of **X = [A; B; C; D; ...]** where **A = [Xa Ya Za]**, etc. **P** is the height of the pole. **F** is the magnitude of the forces in each cable (they are the same in this example). Output **M** is the bending moment around the foot of the pole. Test case: for **X = [20 -40 0; 40 20 0; -10 40 0; -40 -10 0]**, **P = 40**, answer should be **M = [-6044.8 4147.6 0]**.

```

function [M, force] = guideline (X, P, F)
N=size(X); N=N(1); % N is the number of cables extracted from the input matrix
% this turns X into position vectors for each cable PA, PB, etc.
X(:,3) = X(:,3) - P*ones(N,1);
% this begins a loop of converting each position vector into a unit vector:
for i = 1:N
    U(i,:)=X(i,:)/norm(X(i,:));
end;
force = F*U
% cross product of position vector OP and resultant force vector:
M = cross([0 0 P], sum(force));
end

```