

## Numerical Methods for Engineering Technology Students

Gregory K. Watkins

William States Lee College of Engineering  
The University of North Carolina at Charlotte  
Charlotte, NC 28223

### Abstract

Numerical methods is defined as techniques by which mathematical problems are formulated so they may be solved with arithmetic operations. Many of these techniques have great importance in the development of finite element theory and other advanced topics, but are of questionable importance for Engineering Technology students. But if an alternative definition is considered, such as “computer solution of engineering problems,” the subject becomes much more appropriate for an ET curriculum.

In recent years, the personal computer has become as ubiquitous as the television. Today’s students are completely at ease using a computer for everything from researching a term paper to synching data from their PDA to creating CAD drawings. Using the computer to solve engineering problems is not just an easy concept for today’s students to grasp; it’s an expectation.

Concurrent with the rise of the personal computer, and its unprecedented computing power, is the proliferation of software tools available for solving engineering problems. Today, there are numerous commercial packages that an engineering technologist may see on the job. Preparing the student to use these computer based tools should be an important part of ET curricula. A numerical methods course, with the correct focus, can meet this need for today’s student.

UNC Charlotte’s course “Applied Numerical Methods” has recently been updated to include extensive work in Microsoft® Excel and Matlab®, two software packages commonly used by today’s practicing engineers. The course teaches the basic concepts of formulating engineering problems for computer solution. Each topic includes computer based assignments that demonstrate each program’s strengths and weaknesses in that area. Emphasis is placed on engineering applications from mechanical, civil, and electrical disciplines. Students complete the course with two basic competencies: an understanding of how engineering problems are formulated for computer solution, and reasonable proficiency in two commonly used computer based tools.

## Introduction

In recent years, the personal computer has become almost as ubiquitous as the television. Today's students are completely at ease using a computer for everything from researching a term paper to synching data from their PDA to creating CAD drawings. Using the computer to solve engineering problems is not just an easy concept for today's students to grasp, it's an expectation (1). In fact, an issue that frequently arises while teaching engineering to this generation of students is their attitude of "Why should I learn how to do this; the computer is going to solve the problem for me."

Concurrent with the rise of the personal computer, and its unprecedented computing power, is the proliferation of software tools available for solving engineering problems. Today, there are numerous commercial packages that an engineering technologist may run into on the job, and more that are being developed all the time (2). Preparing the student to use these computer based tools should be an important part of today's ET curricula (3). A numerical methods course, with the correct focus, can meet this need for today's student.

Based on an informal survey of institution websites, a course in numerical methods is a common element in many engineering science programs, most often appearing in the mechanical discipline. It is often a graduate level topic, but numerical methods courses do appear in undergraduate curricula as well. The topic, at least by this name, is not nearly as common in engineering technology (ET) programs. The reasons for this are multifaceted, but likely are centered on the subject matter being more theoretical and less applied in nature.

### An ET Course in Numerical Methods

At its essence, numerical methods is defined as "techniques by which mathematical problems are formulated so they may be solved with arithmetic operations (4)." Many of these techniques have great importance in engineering, and are vital in the development of finite element theory and other "high end" topics. However, it's difficult to argue the importance of this type of theoretical information for ET students. But if an alternative definition of numerical methods is considered, such as "computer solution of engineering problems," the subject becomes much more appropriate for an ET curriculum.

Use of commercial software to solve engineering problems has become common in the workplace (2). Spreadsheet programs, such as Microsoft Excel™, now do much more than simply add rows and columns. Excel supports many advanced computational features useful in solving engineering problems such as root finding, simultaneous equations, curve fits, and even optimization. In addition, several technical computing software suites such as Matlab™, Mathcad™, and Mathematica™ are now commonly found in the workplace. Finally, software to perform advanced analysis techniques such as finite element analysis (FEA) and computational fluid dynamics (CFD) has found its way onto desktop computers, when not long ago it was available only in government research labs running on supercomputers (5).

Although a course in CFD software is not likely to be a technical elective in ET programs anytime soon, exposure to commercial software typically found in the engineering workplace

should occur sometime during the student's coursework (3). In addition, at least a basic understanding of some numerical principles, such as how a computer can be programmed with the method of Gauss to solve simultaneous equations, is appropriate material for an ET student. Additional reasons to consider a course in numerical methods are (4):

- In using commercial packages, familiarity with the underlying principles and operations is essential to intelligent use of the programs.
- Using computers to solve otherwise intractable problems increases one's basic proficiency at problem solving.
- Computer tools such as simple programming can be used to extend the capabilities of commercial packages.
- Numerical methods illustrate the power, and limitations, of computers.
- Understanding how advanced mathematical techniques, such as differentiation and integration, are reduced to simple arithmetic operations can actually reinforce the understanding of the original mathematical concept.

#### Numerical Methods at UNC Charlotte

"Applied Numerical Methods" has been a required course in UNC Charlotte's Mechanical and Civil Engineering Technology programs for many years. It has also been available as a technical elective to Electrical Engineering Technology students. The course was initially developed in response to the rise of computer use in engineering problem solving. It carried as a prerequisite a general course in computer programming, and covered the basic concepts involved in using the computer to solve engineering problems. General topics of study included root finding, simultaneous equations, optimization, curve fitting, and numerical differentiation and integration.

The course recently expanded from 2 to 3 credit hours as a result of redefined general education requirements at the university. The additional credit hour offered the opportunity to examine existing content, and to develop additional content that would bring the course in line with current trends. Weaknesses in the existing course were:

- Little reinforcement of concepts
- Tenuous linkage to real-world engineering problems
- No exposure to commercial software

With these weaknesses in mind, the multidisciplinary course has been updated to include two software packages commonly used by today's practicing engineers, Excel, by Microsoft, Inc., and Matlab, by The MathWorks, Inc. The basic concepts, such as root finding, simultaneous equations, optimization, curve fitting, and numerical differentiation and integration, are covered as before, in traditional methods. In addition, each topic is now augmented with lecture material and demonstrations on Excel and Matlab's capabilities in solving those classes of problems. Project assignments, solved on the computer, emphasize real-world engineering applications, as well as the strengths and weaknesses of the programs in each application area.

## Excel & Matlab

Excel is a spreadsheet program by Microsoft, Inc. Spreadsheet programs allow computations on rows and columns of data. The computations are carried out instantly, making spreadsheets an ideal “what if” tool. Excel also includes tools for data visualization such as graphs and charts. Excel has many additional built in tools, such as *Solver*, *Goal Seek*, and the ability to add *Trend Lines*. Excel can also be augmented with *VBA Macros*, a programming tool. A screen shot of Excel’s basic interface is shown in Figure 1.

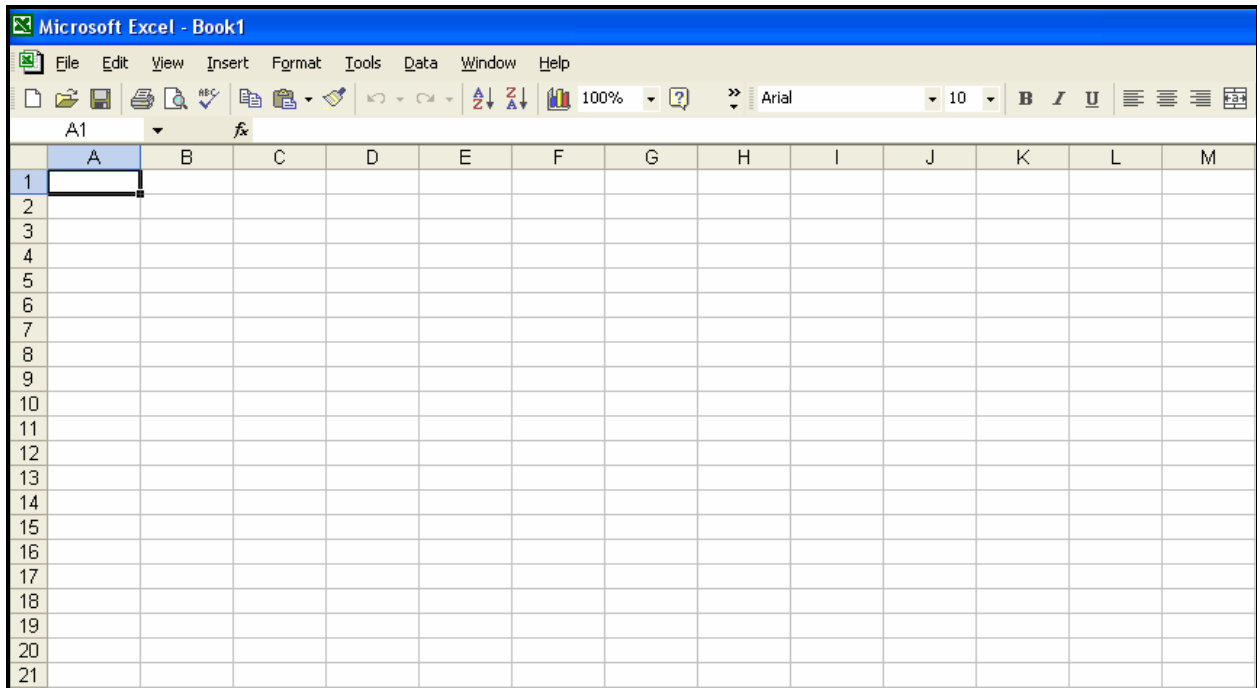


Figure 1 – Screen Shot of Excel Interface

Matlab was originally developed as a *Matrix Laboratory*, and matrices are still a large part of its architecture. Matlab’s interface is command driven and includes *Workspace* variables and a *Command History*. It has numerous built in functions for basic and advanced computations and supports 2-D line plots and as well as 3-D surface plots. Matlab’s capabilities can be extended via *M-files*, which are text files that store program code. Graphical User Interfaces (GUIs) can also be created for custom applications. A screen shot of Matlab’s basic interface is shown in Figure 2.

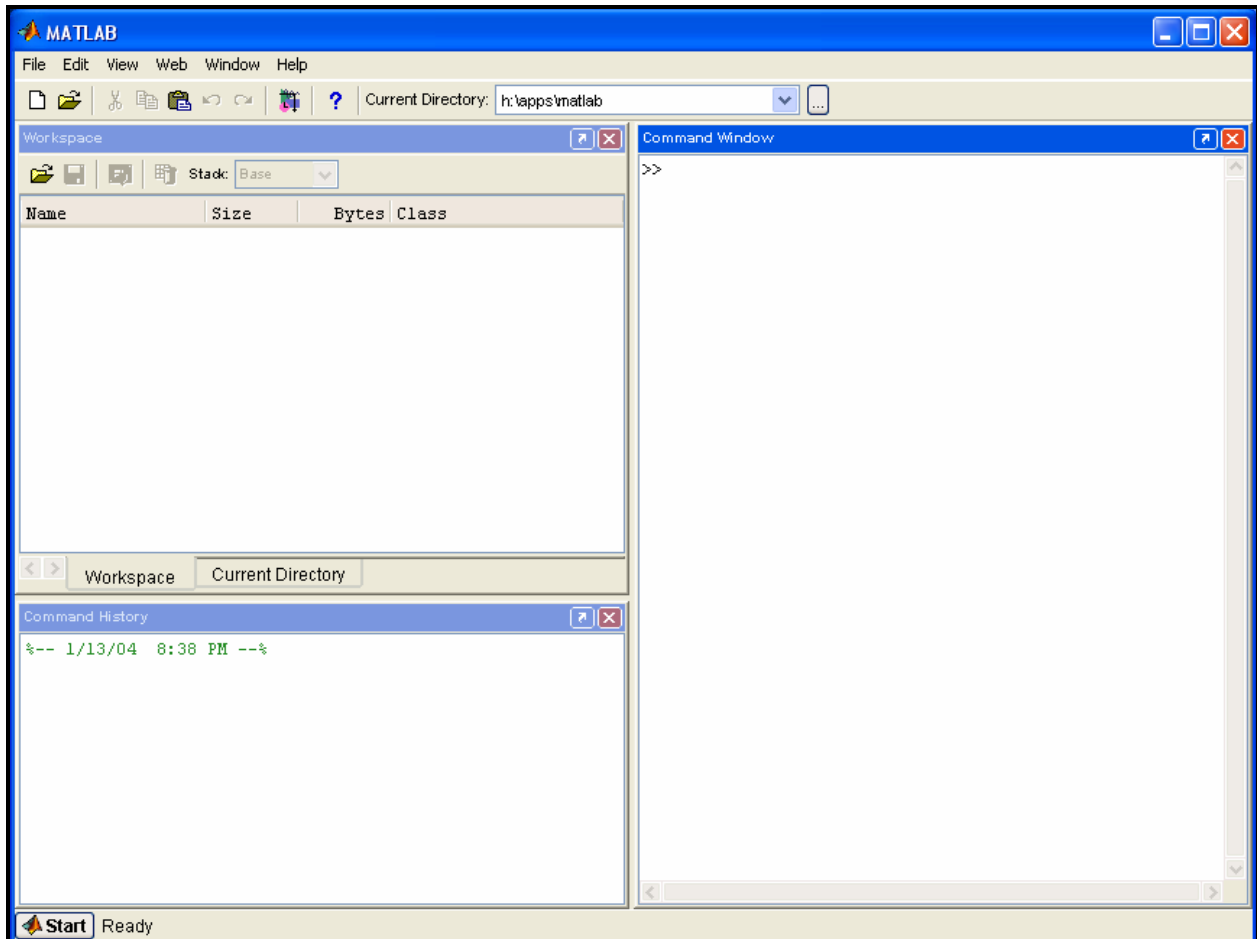


Figure 2 – Screen Shot of Matlab Interface

Neither program is all encompassing and each has different strengths and weaknesses. Exposure and some competency in both programs greatly increases the range of problems a student can address.

### Course Content

As mentioned earlier, the numerical methods course at UNC Charlotte covers the basic topics of root finding, simultaneous equations, optimization, curve fitting, and numerical differentiation and integration. Each topic is first covered with traditional lectures and homework assignments. Most homework assignments are completed with pencil, paper, and calculator. The goal here is to give the student a clear understanding of the underlying concepts. Tedium is kept to a minimum, though is at times unavoidable, such as performing row operations in the method of Gauss. Only after the concepts have been enforced in this manner are the computer tools introduced.

With a (hopefully) clear understanding of the topic established, the focus then turns towards Excel and Matlab's capabilities in solving that particular class of problem. Lecture material

covers the basics, then a sample problem is solved on a computer overhead. Handouts, including screen captures, are provided that take the students step by step through a similar sample problem. Finally, a comprehensive real-world application problem is assigned for computer solution, one that is typically infeasible to solve by hand.

In the following sections, each numerical method topic is summarized along with the corresponding project assignments. Screen shots of output from selected projects are included in the appendix. The project handouts are freely available in PDF format on the course website at [www.coe.uncc.edu/~gkwatkin/ETGR3272/](http://www.coe.uncc.edu/~gkwatkin/ETGR3272/). They are also available in Word™ format by email request to [gkwatkin@uncc.edu](mailto:gkwatkin@uncc.edu).

i. Modeling, Computers, and Error Analysis

The course begins with an introduction to common methods used in posing engineering problems for computer solution. Computer basics, such as bits, bytes, binary representation of numbers, and error analysis are introduced. Programming concepts, such as code structure, do loops, and subroutines are reviewed. The following computer projects are assigned, which are based on modeling a falling parachutist.

- Project 1 Excel project covering the basics of data entry, equations, and plotting.
- Project 2 Matlab project that covers interactive use, plotting, and programming with M-files.

ii. Root Finding

The first true numerical method covered is root finding. Applications that produce equations that cannot be solved algebraically are used to motivate the concept. Four different techniques are covered: Bisection, False Position, Newton, and Secant. Roots of polynomials are also discussed. The following computer projects are assigned, which include applications in statics, electrical circuits, plane geometry, and projectile motion:

- Project 3 Excel project that applies the *Goal Seek* tool to root finding. Also introduces the equation editor.
- Project 4 Matlab project that examines functions related to problems of root finding. Also introduces user defined functions and polynomial operations.

iii. Simultaneous Equations

This section begins with a review of matrices, including mathematical operations such as addition, multiplication, and row operations. Simultaneous equations and their applications are introduced, including representation in matrix form. Solution techniques include Gaussian Elimination with pivoting, the iterative technique Gauss-Seidel, and solution by use of the matrix inverse. In addition, plotting of 3-D surfaces is reviewed in

the context of solving three simultaneous equations, with the analogy drawn to intersecting lines representing the solution of two simultaneous equations. The following computer projects are assigned, which include applications in multi-channel flow, static friction, building construction, and electrical circuits:

- Project 5 Excel project that illustrates the solution of simultaneous equations. Introduces the matrix functions MINVERSE and MMULT.
- Project 6 Matlab project that illustrates the solution of simultaneous equations. Also introduces Matlab's utility in plotting 3-D surfaces.

iv. Optimization

Though a very complex topic, the underlying concept of optimization is actually quite simple. The topic is motivated through review of the calculus technique of taking the first derivative and setting it equal to zero. Hand problems are limited to one-dimensional algorithms, such as the Golden Section Search, Quadratic Interpolation, and Newton's Method. More robust algorithms are introduced via computer assignments. The following computer projects are assigned, which include applications in growth rate, electrical circuits, airfoil drag, and pollution concentration:

- Project 7 Excel project that introduces optimization techniques using the *Solver* tool.
- Project 8 Matlab project that introduces functions specifically designed for optimization.

v. Curve Fitting

Curve fitting is possibly one of the most useful numerical techniques covered in the course. Curve fits via linear, polynomial, and least squares regression are covered. In addition, the following interpolation techniques are introduced: Newton's Divided-Difference, Lagrange Polynomials, and Splines. The following computer projects are assigned, which include applications in analysis of experimental data:

- Project 9 Excel project that demonstrates curve fitting techniques available in Excel, which are *Trend Lines* added to an existing plot.
- Project 10 Matlab project that introduces curve fitting capabilities. Makes use of the *Basic Fitting Interface*, which is a GUI (Graphical User Interface) created within Matlab.

vi. Numerical Differentiation and Integration

Numerical differentiation and integration are motivated primarily by illustrating computation of definite integrals where the function cannot be integrated by hand. Finite

difference techniques are introduced in the context of numerical differentiation. The following computer project is assigned, which includes applications in heat transfer and electrical circuits

- Project 11 Matlab project that introduces the program's capabilities in the areas of numerical integration and differentiation.

### Conclusion

A course that covers the more common and powerful numerical method techniques is appropriate for engineering technology students at the baccalaureate level. Typical subjects include root finding, solution of simultaneous equations, optimization, curve fitting, and numerical differentiation/integration. Such a course should also expose the student to commercial software that can illustrate the techniques, as well as solve problems that cannot reasonably be solved by hand. The software should also be representative of that which may be expected in the workplace. Excel and Matlab are two suggested packages.

### References

1. Clewley, R., "Old-Schooler Teaches New Tricks," *Wired Magazine*, September 2001
2. Hodge, B. K., Taylor, R. P., "Factors for Change in Mechanical Engineering Education," Proceedings of the 1997 ASEE Annual Conference and Exhibition, Session 2266, Milwaukee, WI, June
3. Criteria for Accrediting Engineering Technology Programs, Conventional Criteria I.C.6., Technology Accreditation Commission Accreditation Board for Engineering and Technology, Inc. 111 Market Place, Suite 1050 Baltimore, MD 21202 Telephone: 410-347-7700
4. Chapra, S. & Canale, R. (2002) *Numerical Methods for Engineers*, McGraw-Hill.
5. Baker, A.J., Pionke, C. D., Taylor, M. J., and Luttrell, B., 1996, "FEM Brought to the Academic Engineering Desktop," Proceedings of the 1996 ASEE Southeastern Section Meeting, Gatlinburg, TN, April

### Biography

#### GREGORY K. WATKINS

Gregory Watkins received a B.S. in Mechanical Engineering from North Carolina State University, a Master of Engineering Management from Old Dominion University, and a Ph.D. in Mechanical Engineering from UNC Charlotte. He has taught in the Engineering Technology department at UNC Charlotte for the past 1.5 years. He taught in the Engineering Technologies Division at Central Piedmont Community College for 8 years and has 9 years of industrial work experience.



Appendix

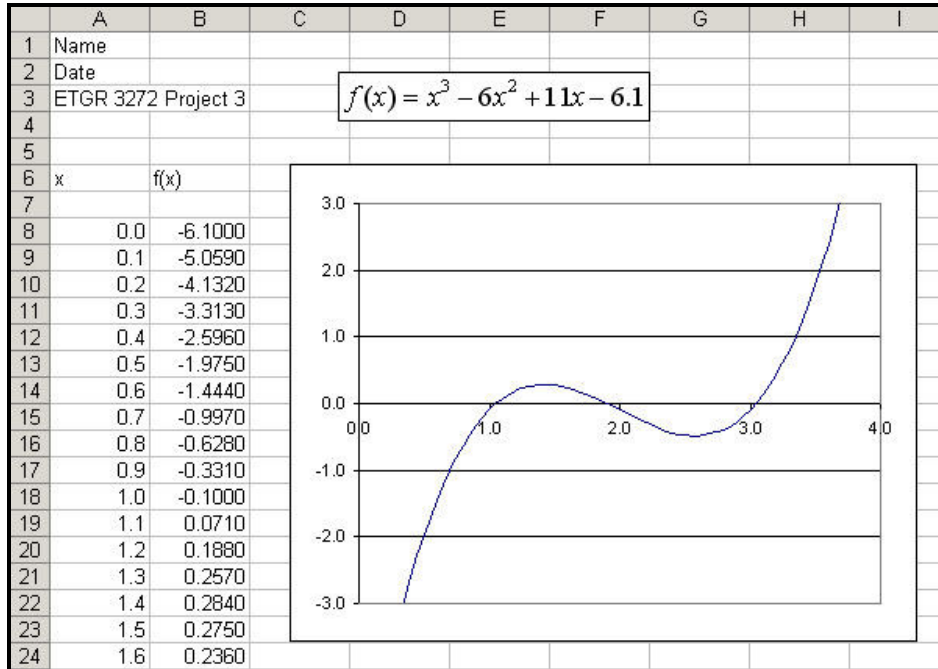


Figure A1 – Sample Output from Project 3 (Excel)

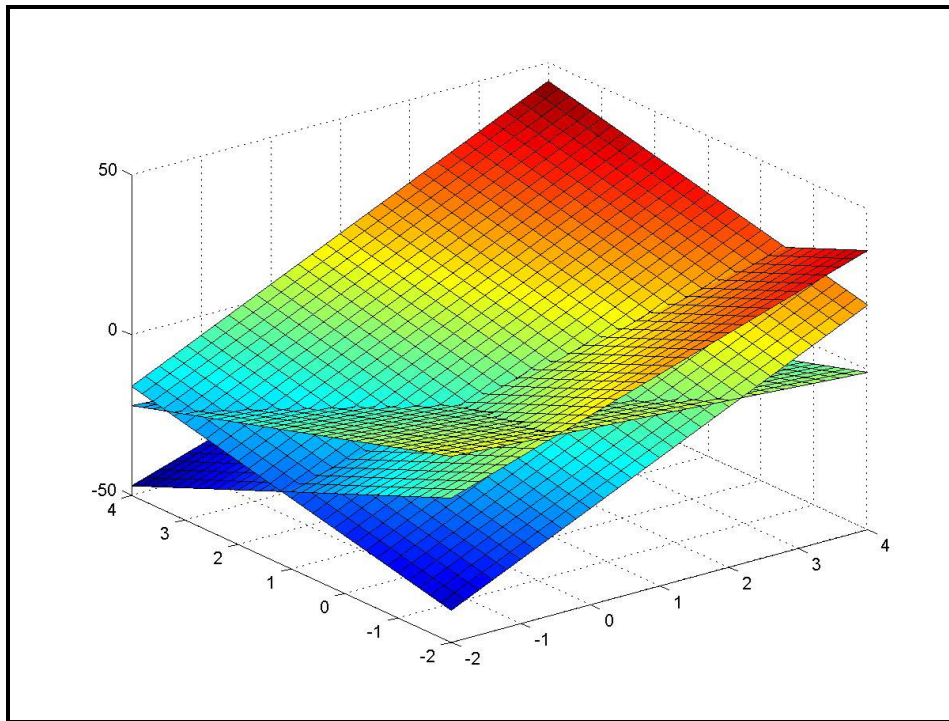


Figure A2 – Sample Output from Project 6 (Matlab)

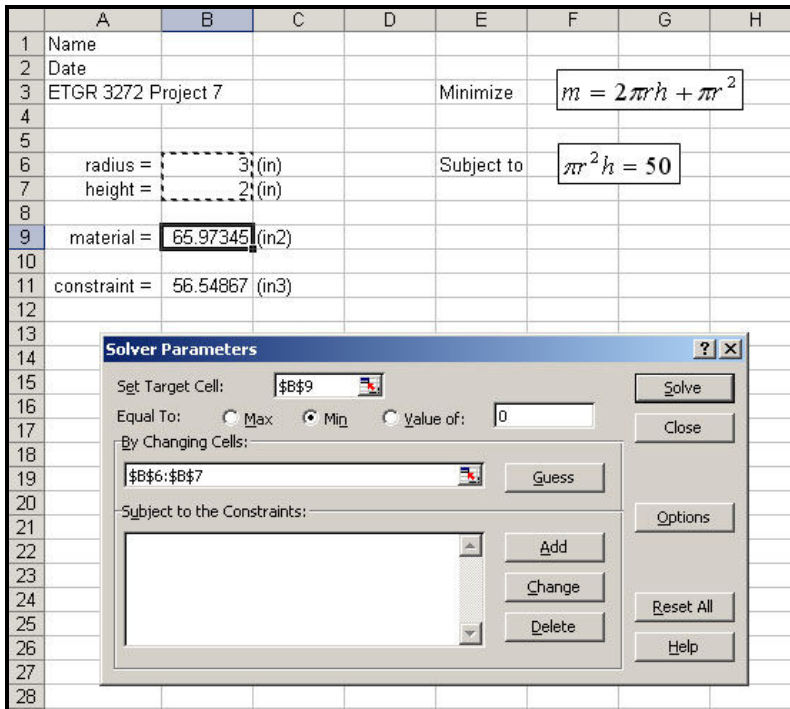


Figure A3 – Sample Output from Project 7 (Excel)

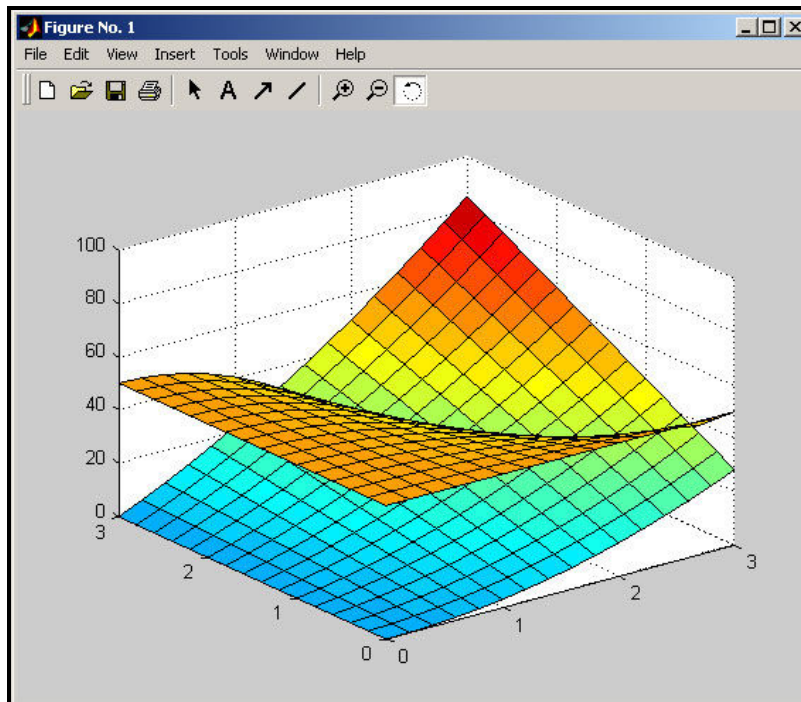


Figure A4 – Sample Output from Project 8 (Matlab)

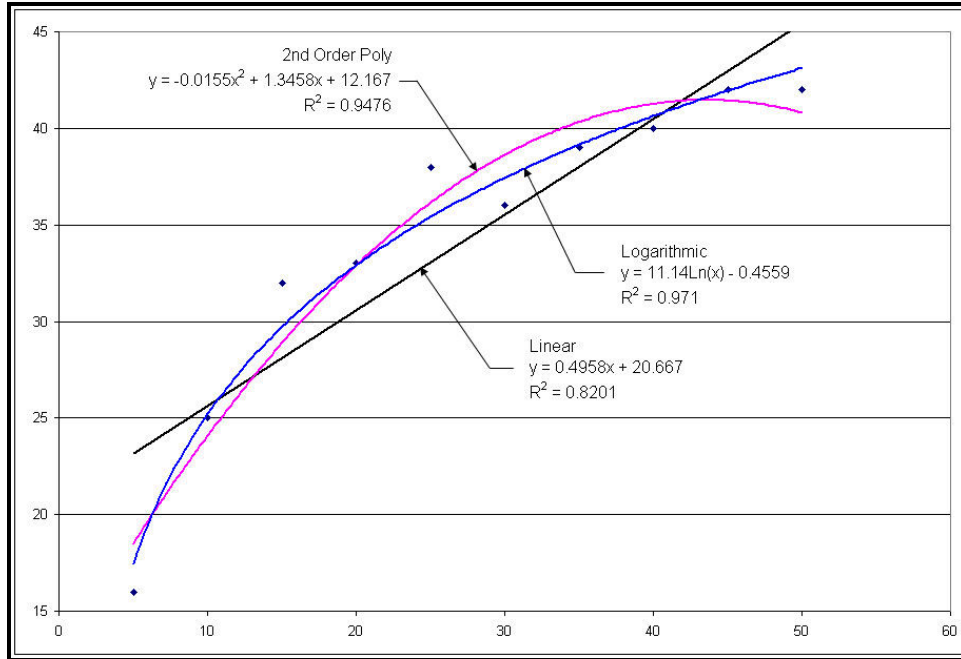


Figure A5 – Sample Output from Project 9 (Excel)

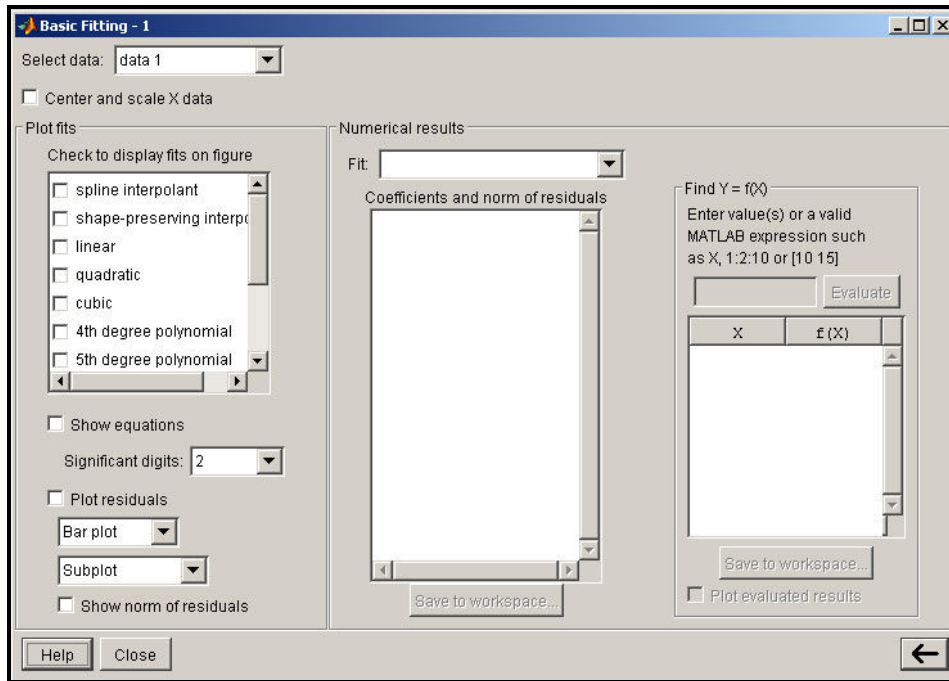


Figure A6 – Sample Output from Project 10 (Matlab)