2006-87: USING MACROS AND FUNCTIONS TO ENHANCE TEACHING NUMERICAL METHODS TO ENGINEERING STUDENTS

Jonathan Lambright, Savannah State University

Using Macros and Functions To Enhance Teaching Numerical Methods To Engineering Students

Abstract

Using mathematical models to predict the behavior of physical systems is an important part of engineering analysis and the techniques of numerical methods are important tools that are used to facilitate the mathematical modeling process. This paper describes techniques developed and used in a numerical methods classroom for engineers titled “Computational Modeling”. The course teaches the foundations of development and usage of numerical methods such as Euler’s method, the Newton-Raphson method, and elimination methods. The techniques used in the classroom are those of developing Microsoft Excel macros and Matlab M-file functions to program the numerical methods and subsequently solve mathematical models of the physical world.

There are two important aspects of using these techniques in the Numerical Methods classroom. First, numerical methods are computationally intensive and structured computer programs provide a most efficient way to solve the mathematical models. Many of the methods are iterative in nature and are facilitated through the use of computer programs. By using the macros and functions it helps to portray to the students the difference between using two or more different numerical techniques and the types and magnitudes of errors that may result. The second main reason for using macros and m-file functions in the numerical methods classroom is that it reinforces the need for engineers to develop well written and structured computer programs that are used as tools in solving engineering-based problems. This paper details the different types of numerical methods that were taught in the class. The paper describes the macros and m-files that were developed as supplemental teaching tools, how they were used in the classroom and how it benefited the instruction of the engineering student.

Introduction

Numerical methods are used in approximating derivatives and solving nonlinear equations. Many differential equations cannot be solved analytically, in which case we have to satisfy ourselves with an approximation to the solution. Using mathematical models to predict the behavior of physical system is an important part of engineering analysis. In addition, the techniques of numerical methods are important tools that are used to facilitate the mathematical modeling process. The instructional techniques and algorithms described in this paper were used to compute such an approximation and facilitate teaching numerical techniques to students. Alternative methods are to use techniques from calculus to obtain a series expansion of the solution. The techniques used in the classroom are those of developing Microsoft Excel macros and Matlab M-files functions to program the numerical methods and subsequently solve the mathematical models of the physical world. Using Matlab and Excel facilitate demonstrating the need for repetitive analysis in numerical techniques.

The class where the instructional techniques were used was an introductory class to numerical methods. The types of numerical methods techniques studied included Taylor series (backwards difference, centered difference, forward difference), Euler's method, Open methods (fixed-point iteration, Newton-Raphson, secant method), Bracketing methods (graphical method, bisection method, false-position method) and Gauss Elimination/ Cramer's Rule. Samples of some of the methods emphasized during the course are:

Euler method .....

$y(t_{i+1}) = y(t_i) + f(t_i, y_i)(t_{i+1} - t_i)$ or new value = old value + slope x step size.

where $f(t_i, y_i)$ represents the slope or derivative at (t_i, y_i) and $(t_{i+1} - t_i)$ is the step size.

Taylor series....

Backwards difference: $f'(x_i) = \dfrac{f(x_i) - f(x_{i-1})}{h}$

Center difference: $f'(x_i) = \dfrac{f(x_{i+1}) - f(x_{-1})}{2h}$

Forward difference: $f'(x_i) = \dfrac{f'(x_{i+1}) - f(x_i)}{h}$

h = step size $(x_{i+1} - x_i)$

Bracket methods...

False position: $x_r = x_u - \dfrac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$

where x_l is the lower guess and x_u is the upper guess.

The course began the study of numerical methods techniques by analyzing the Euler method. Euler's method was used to demonstrate approximating the solution to its initial value. The Euler method is not complex and is a good start to show how to solve the first-order initial value problem. Euler's method can be intuitively derived from several different avenues and, for this reason, is often considered to be the cornerstone method on which the framework to formulate and discuss subsequent methods is built. Euler's method can be derived using the given ordinary differential equation and Taylor polynomials.

Why Matlab Functions and Excel Macros

Many of the numerical methods techniques are iterative in nature. The process of finding an initial value and substituting that value back into the numerical approximation is inherent in numerical methods. Many numerical analyses require several iterations and would be laborious if completed by hand. Structured programming languages and computers make employing these methods much more convenient and productive. Structured programs implemented in such applications such as Matlab and Excel are ideal for solving numerical approximations because of the need for repetitive looping and conditional termination. Students can also compare and contrast each method, observing the magnitude of errors that may result from each. Engineers use structured programs to solve real-world engineering-based problems.

Applying programming skills gives students the reinforcement that engineering demands. (Versatile, well-rounded, enriched)--especially in the more general engineering field (mechanical).

Instructional Approach

Computing Techniques and Prerequisites

The "Computing Techniques" class was taught during the fall semester of the 2005 – 2006 academic year. In teaching numerical methods, students were expected to have had a background in programming such as MATLAB and be familiar with Microsoft applications such as Microsoft Excel/Visual Basic. These tools were the foundation for the instructional approach. The students therefore should have taken the prerequisite class "Introduction To Computing For Engineers and Scientists" or an equivalent class that taught the concepts of computing including procedural and object oriented programming. The students had to be comfortable with developing multiple interrelated functions with multi-argument input. The students also had to be familiar with looping and termination conditions. The choice to use Matlab was decided because of the fact that the students were familiar with the software and were already used to developing procedural based conditional programs that required looping and recursion. Extending the tools to the usage of macros in Excel was not difficult as the students only had to learn the syntax of the language. The MATLAB m-files and Microsoft Excel macros assisted in giving the students proficiency in programming skills.

Structure of the Class

The projects assigned in the computing techniques class were a significant portion of the students grade. Other significant parts of the students grade came from in classroom examinations during the semester and one final examination at the end of the semester. The in class room examinations were given to make sure that the students understood the fundamental concepts of each numerical method. Students were tested on open and bracketed numerical techniques such as Fixed Point and False Position methods, however on exams and hand written home works the students were asked to perform the technique to within a percentage error that usually required no more than 4 to 5 iterations.

Purpose of Programming Learned Numerical Techniques

After the students studied the fundamentals of a particular numerical method they were asked to solve a problem that required iterations that brought the percent relative error to a level that would require much more than 4 or 5 repetitions. Hence, the students quickly see the need for and power of computers and procedural language tools such as MATLAB and Excel macros in solving numerical method problems. For example, if the students were asked to solve these problems by hand, it would have been a very laborious task and would have quickly undermined the spirit of the student. Such problems teach our students the value of powerful tools and how they can assist us in solving real world engineering problems.

Two Programmed Numerical Methods Examples

Below is the result of one numerical method application that was programmed by one group of students. For this particular lab, the students were divided into 3 separate groups. One group had to develop a MATLAB application to perform the Taylor Series Forward Difference approximation. One group had to develop a MATLAB application to perform the Taylor Series Centered Difference approximation. One group had to develop a MATLAB application to perform the Taylor Series Backward Difference approximation. The students had to differentiate the same polynomial as shown below:

$$f(x) = -0.1 * x^4 - 0.15 * x^3 - 0.7 * x^2 - 0.5*x + 2$$

****

% Taylor Series Centered Difference Approximation in MATLAB script form.

h = input (‘please enter the desired step size:’)
%this is the step size; the smaller the step size the more accurate the result

x = input (‘please enter a value for x:’)
%this is the initial value (starting point)

d = -0.4 * (x ^3) – 0.45 * (x ^2) – 0.14 * (x) – (0.5)
% This is the first derivative of the polynomial

f = -0.1 * (x^4) – 0.15 * (x ^ 3) – 0.7 * (x ^2) – (0.5 * x) + 2
%This is the original polynomial

y = x – h
z = x + h
%here, the function is split into two components y and z…it calculates each separately then merges them

g = -0.1 * (y^4) – 0.15 * (y^3) – 0.7 * (y ^2) – (0.5 * y) + 2
%this is the value when y is substituted into the equation (for x – 1)

i = -0.1 * (z ^4) – 0.15*(z ^3) – 0.7*(z ^2) – (0.5 * z) + 2
%this is the value when z is substituted into the equation (for x + 1)

center = (i – g)/ (2 * h)
%this computes the centered difference; the variables i, g, and h were used to make this computation modular and easy to visualize.

truncation = abs (((d – center)/d) * 100)
%this computes the truncation error

****

Below is another numerical method application that was programmed by each student in the class. For this lab the students had to develop a macro in Excel that solved a differential equation using the Euler method (see Chapra and Canale[2]). The equation representing a falling object is

$$V(t_{i+1}) = v(t_i) + [g - (c \bullet v(t_i)/m)]\ (t_{i+1} - t_i)$$

Where $[g - c \bullet v(t_i)/m]$ represents the derivative or slope and $(t_{i+1} - t_i)$ is the step size.

****

```
Function Euler (dt, ti, tf, yi, m, cd)

Dim h As Single, t As Single, y As Single, dydt As Single
t = ti              / the initial time
y = yi              / an initial value of y
h = dt              / the step size
Do
        If t + dt > tf Then
         H = tf - t
        End If
        dydt = dy (t, y, m, cd)         /compute the derivative or slope
        y = y + dydt * h                / The Euler method
        t = t + h                       /increase the time
        If t >= tf Then Exit Do         / an exit condition
Loop
Euler = y
End Function


Function dy( t, v, m, cd)               / a function to compute the derivative
Const g As Single = 9.8
dy = g – (cd / m) * v                   / this result is input into the equation in the Euler function
End Function
```

****

Benefits of Using Excel Macros and MATLAB Functions

The benefits of using tools such as MATLAB to teach numerical methods are many. First the students are not just taught the rigor of crunching through numerical approximations but they clearly see the reason why we would want to use procedural programming tools in such numerical techniques. The author typically starts the introduction of a numerical method by solving a given problem by hand. After about 5 or six iterations, students are quick to see the value of using computers and programming as tools to help us solve engineering problems. Next the students programming skills are sharpened and honed as they build applications to tackle numerical approximations of real world systems. Finally, and one of the biggest payoffs is that the procedural program allows for variables such as the step size and initial value to be repeatedly and rapidly changed; thus producing different outcomes based upon the input values. The beauty of this is that it enhances the students learning process in that they can witness, in a very short period of time, the effects of reducing variables such as the step size upon the accuracy of the solution and the number of iterations needed to converge to a specified tolerance range. To accomplish this lesson by hand without these powerful tools would take up valuable class time.

Future Work

There still remains a fair portion of work that must be done in order to realize the vision that is hoped for in the “Computing Techniques” course. It is envisioned that the instructor and students will have at their disposal an analytical tool that will allow them to use numerical analysis to solve differential equations. The most powerful portion of such a tool in the classroom would be the ability to modify variables and watch the effects upon outcomes such as truncation error and percent relative error. Tools such as Matlab and Excel will be used to help create the end product. The intent is not to recreate these tools but to use them to develop an environment that will enhance the learning capability in the numerical methods classroom.

Conclusions

Numerical analysis is a very important part of solving real world engineering problems. Numerical Methods such as the Taylor Series, False Position, and the Bisection method are invaluable tools that assist in the creating and solving approximations to non linear equations. Students must be taught the fundamentals of developing and solving these numerical approximations by hand. However, theory must be combined with technology and hands on practice to emphasize the need for tools such as Matlab and Excel in solving engineering problems through numerical approximations. By implementing such tools in the classroom, students sharpen their programming and analytical thinking skills. In addition, students can experience the need for and the power of these tools in solving real world problems and use the experience to creatively think of newer ways to solve engineering problems.

References

[1] Hanselman, D., and Littlefield, B., “Mastering MATLAB 7: A Comprehensive Tutorial and Reference”, Prentice Hall Publishers.

[2] Chapra, S.C., and Canale, R.P., “Numerical Methods For Engineers”, Fifth Edition, McGraw Hill Publishers.

[3] Stanley, William, D., “Technical Analysis and Applications with MATLAB”, Thomson Delmar Learning Publishers.