California State University Northridge

College of Engineering and Computer Science
Mechanical Engineering Department

Mechanical Engineering 309
Numerical Analysis of engineering Systems

Spring 2014 Number: 15237 Instructor: Larry Caretto

Sample Programming Exam (2 hours)

INSTRUCTIONS: Write a VBA or MATLAB program to accomplish the tasks outlined below. At the end of the quiz email me your work (Excel workbook or MATLAB file with both function listing and command history to execute the function and display the results.) You can use your textbook, any class notes and the online VBA or MATLAB help systems. You must do this assignment with no help from anyone in or out of this class. **You cannot use the internet to find code.** You have to do this quiz with no help from the instructor. You can ask for help if you are not certain about what is required or if have a severe problem such as a malfunctioning computer.

Write a VBA or MATLAB function that repeatedly uses Simpson's rule (cutting the step size, h, in half for each repitition) to evaluate an integral numerically until the relative differences between two evaluations of the integral is less-than or equal to some user-specified value. Simpson's Rule for estimating the integral $I = \int_a^b f(x)dx$ is shown below.

$$S = \frac{h}{3}\left[f_0 + f_N + 4\sum_{k=1,3,5}^{N-1} f_k + 2\sum_{k=2,4,6}^{N-2} f_k \right] \qquad I = S + O\left(h^4\right)$$

In this equation the interval [a b] is subdivided into N panels of width h = (b – a)/N, where N is an even number. The following definitions apply: $f_k = f(x_k)$ where $x_k = a + kh$.

Algorithm

Your code should start by evaluating S with N = 2. It should then have a loop where it repeatedly doubles N and recomputes S. The loop should terminate when two successive values of S are close to within a specified maximum relative error or when a specified maximum number of iterations are exceeded. Your function should return either a value of S (if converged) or an error message (if the allowed iterations were exceeded.) You should write a separate function for the integrand, f(x). A user should not have to make any changes in your Simpson's rule subroutine to evaluate a new integral.

The basic approach outlined below is simpler to apply and will count for 80% of the maximum exam grade. The advanced approach, if successful, will count for 100% of the maximum grade.

Basic approach: initially, write the code where you compute the integrand for all values of x_k at each step size and apply it to the test problem in the "What is required" section below.

Advanced approach: You may start with this approach or modify your basic-approach code to accomplish this work. In this advanced approach, for each new value of N, the code only has to compute the integrand for new values of x_k (those for which k is odd). To do this consider the Simpson rule to have three terms, the end terms, $f_0 + f_N$, the even-k terms (terms for k = 2, 4, …, N-2, which are multiplied by 2) and the new, odd-k terms (terms for k = 1, 3, …, N – 1, which are multiplied by 4.) As you double N for new iterations: (1) the end terms do not change; (2) the sum for the even-k terms at the new step can be found as the sum of the even-k and odd-k terms at the previous step as shown in the equation below. You only have to compute the new odd sum for k = 1 to N – 1.

$$\left(\sum_{k=2,4,6}^{N-2} f_k \right)_{h=(b-a)/N} = \left(\sum_{k=1,3,5}^{N/2-1} f_k + \sum_{k=2,4,6}^{N/2-2} f_k \right)_{h=2(b-a)/N}$$

What is required

Submit an Excel workbook (or a MATLAB file that contains both the program listing and the command window results). Evaluate the integral of $e^{\alpha x}\sin(\beta x)$ from 0 to 1 for $\alpha = 0.3$ and $\beta = 3$. Use a relative error of 10^{-10}. Compare your result to the exact value $\int e^{\alpha x} sin(\beta x) dx = \frac{e^{\alpha x}[\alpha sin(\beta x) - \beta cos(\beta x)]}{\alpha^2 + \beta^2}$. If you cannot complete the assignment, write a statement of the next steps you would take in your program.