Homework 1: Numerical Integration, differentiation, and bisection

A.L. Phillips
Department of Physics,Rensselaer Polytechnic Institute
philla3@rpi.edu

12 February 2013

Chapter 1

Integration, differentiation, and bisection

1.1 Background

1. To date,Mathematical operations may be conducted analytically or numerically. Analyti....

1.2 Sources of Error

1. Roundoff vs Truncation
   Machine epsilon... Two sources of error in numerical methods are Roundoff and Truncation errors. Round-off errors are the result of systems having a finite quantity of significant figures to represent numbers. For example, if a computer is capable of can store three significant figures, then it could approximate $\frac{1}{3}$ as 0.333. Because the computer is not representing $\frac{1}{3}$ *exactly* as $0.\overline{3}$, a round-off error (ξ_{max_T}) occurs.
   In this case, $\xi_{max_T} = \dfrac{1}{3} - 0.333 = 0.000\overline{3}$

   Truncation error is the result of truncating (shortening) a mathematical procedure. In numerical methods, the shortening of a mathematical procedure occurs whenever we are required to approach zero (integrate) or use an infinite number of terms (series). For example, the Maclaurin series expansion of $cos(x) = 1 - \frac{1}{2}x^2 + \frac{1}{24}x^4 - \frac{1}{720}x^6 + \ldots$ .Since we must chose a finite number of terms, a truncation error will ultimately result. If we choose two terms, the truncation error would then equal the sum of all the excluded terms. Namely, $\frac{1}{24}x^4 - \frac{1}{720}x^6 + \ldots$.

1.3 Numerical Integration Techniques

1. Trapezoidal Method:

   The trapezoid rule calculates the area under a curve much like a Riemann sum. Here, rather than using rectangles, the trapezoidal method employs trapezoids. The graphic below depicts the use of a single trapezoid to approximate the area under the blue curve. The more trapezoids that are added between a and b, the closer the "secant-shaped" tops of each trapezoid will become to the tangent on the curve above it.
   trapezoid.png

   The general formula for the approximation of a definite integral via the Trapezoid Rule is as follows:
   $$\int_a^b f(x)\,dx \approx \frac{\Delta x}{2}\Big[f(x_0) + 2f(x_1) + \ldots + 2f(x_{n-1}) + f(x_n)\Big]$$
   where $\Delta x = \dfrac{(b-a)}{n}$ .... $x_i = a + i(\dfrac{b-a}{n})$

   Trapezoidal Error:

   The maximum amount of truncation error (ξ_{max_T}) when using the trapezoid rule may be determined as follows:

   $$\xi_{max_T} = \frac{(b-a)^3\left|f^{(2)}_{(max)}\right|}{12n^2}$$

   Here, b and a are the upper and lower limits of integration, n is the trapezoid quantity being used and $f^{(2)}_{(max)}$ is the second derivative of the function. The (max) subscript indicates that the second derivative should be evaluated at the point from $[a, b]$ which maximizes the value of $f^{(2)}$.

   Similarly, the minimum number of trapezoids needed to calculate an integral to within a given accuracy (ξ_{max}) may be determined by solving the above equation for n:

   $$n \geq \sqrt{\frac{(b-a)^3\left|f^{(4)}_{(max)}\right|}{12(\xi_{max})}}$$
   Note: n must be a whole number; therefore if $n \geq 10.5$ then choose $n = 11$

2. Simpson's Rule:

integration-eps-converted-to.pdf

3. Gaussian Quadrature:
4. Comparison:

1.4 Numerical Differentiation Techniques

1. Test

1.5 Algorithmic Considerations

1.6 Implementation

1.7 Results

1.8 Discussion

1.9 References

1.10 Original Assignment

1. Write a double-precision program to integrate an arbitrary function numerically using the trapezoid rule, the Simpson rule, and Gaussian quadrature (for GC, limit yourself to a simple 3 point approximation within each interval: use one interval only in this case). In the discussion of your results, a plot of the relative error $\epsilon = |\frac{\text{numerical}-\text{exact}}{\text{exact}}|$ for each approach should be included.

2. Compute the first derivative of x^2, x^3, e^{-x}, and another well-behaved, non-trivial, function of your choice, using forward difference, central difference, and 5-point approximation.

   (a) In each case plot the error as function of step size (consider using log-log scales)

   (b) In your discussion, make sure to talk about the best step size to use (Refer to the class notes).

   (c) Make sure you consider the particular cases where a given approximation is exact.

3. Write a C++ program that finds the zero of a function using the bisection method.
4. (optional, for 2 extra credits) Write a C++ program that finds the zero of a function using the newton-Raphson method.

The section titles provided below are suggested as guides, feel free to adapt for your own presentation. You do **not** need to provide listings of your program (but you can if it helps the discussion).