Programming Assignment 5: Numerical Integration

CS 2073, Computer Programming with Engineering Applications

Spring Semester, 1992

For this assignment, we want a program that will do numerical integration. You don't really need to know any calculus, since for us the integral of a function will just be the area under its graph, or its average value. This assignment will use three numerical integration methods:

- the trapezoid method,
- Simpson's method, and
- a Monte-Carlo method.

We will be finding the value of the integral of a function $f(x)$, for x from a to b. You will also start with an integer n representing the number of intervals to divide the segment from a to b into. The size of each interval is $h = (b - a)/n$. Given these starting values, the trapezoid method uses the formula

$$\text{Integral} \approx (h/2)\Big[f(a) + 2\cdot f(a+h) + 2\cdot f(a+2h) + 2\cdot f(a+3h) + 2\cdot f(a+4h) + \ldots + 2\cdot f(a+(n-2)h) + 2\cdot f(a+(n-1)h) + f(b)\Big]$$

Similarly, Simpson's method uses the formula

$$\text{Integral} \approx (h/3)\Big[f(a) + 4\cdot f(a+h) + 2\cdot f(a+2h) + 4\cdot f(a+3h) + 2\cdot f(a+4h) + \ldots + 2\cdot f(a+(n-2)h) + 4\cdot f(a+(n-1)h) + f(b)\Big]$$

Finally, a Monte-Carlo method might use

$$\text{Integral} \approx h\cdot\Big[f(x_1) + f(x_2) + f(x_3) + \ldots + f(x_{n-1}) + f(x_n)\Big]$$

Here the numbers $x_1, x_2, \ldots, x_n$ are randomly chosen from the interval from a to b.

You should use the two specific functions

$$f(x) = 1/(1 + x^2), \text{ for } x \text{ from 0 to 1, and}$$

$$g(x) = e^{-x^2}, \text{ for } x \text{ from -2 to 2.}$$

Finally for each of the two functions, and for each of the three integration methods (6 cases), you should try $n = 10$, and $n = 1000$. Thus you should have 12 answers altogether, and your answers should be clearly labeled with the function (f or g above), the values of a and b, the value of n, and the integration method.

You must use Pascal `function`s to calculate f and g as above.

You must use a procedure `generate_f` that will take as inputs the numbers a, b, and n, and will return (as a reference parameter) an array of function values `funcval`, with $f(a)$, $f(a+h)$, $f(a+2h)$, . . . , $f(a+(n-1)h)$, $f(b)$ stored in array locations 0 through n. Similarly for a procedure `generate_g`:

```
const Maxval = 1000;
type funcvaltype = array[0..Maxval] of real;
procedure generate_f (var funcval: funcvaltype; a, b: real; n: integer);
procedure generate_g (var funcval: funcvaltype; a, b: real; n: integer);
```

Another Pascal procedure `generate_fm` should take a, b, and n as inputs and generate n function values $f(x_1)$, $f(x_2)$, $f(x_3)$, . . . , $f(x_{n-1})$, $f(x_n)$ in the array `funcval`, and similarly for a procedure `generate_gm`:

```
procedure generate_fm(var funcval: funcvaltype; a, b: real; n: integer);
procedure generate_gm(var funcval: funcvaltype; a, b: real; n: integer);
```

Then you must have three Pascal functions `trap`, `simp` and `monte` that use the array `funcval` and the values a, b, and n, to calculate the integral according to the above formulas:

```
function trap  (funcval: funcvaltype; a, b: real; n: integer): real;
function simp  (funcval: funcvaltype; a, b: real; n: integer): real;
function monte (funcval: funcvaltype; a, b: real; n: integer): real;
```

The random numbers can be generated using a random number generator that will be separately distributed and discussed in class. Let's use type `double` for all real numbers, rather than `real`. Print answers with 16 significant digits.

(Note: for efficiency sake above, one might want to make the `funcval` parameters to the three functions above reference parameters.)

Extras:

In addition to the above, find an approximate value for the integral of $g(x)$ from $-\infty$ to $+\infty$.