MAT128A: Numerical Analysis, Section 2
Project One: The $(n+1)$-point Trapezoidal Rule
Due Date: October 12, 2018

1. Project description

The integral

$$\int_a^b f(x)\ dx \tag{1}$$

can be (crudely) estimated by approximating the area under the graph of $f(x)$ by a trapezoid. In other words, by approximating the integrand f via the affine function

$$g(x) = \frac{f(b) - f(a)}{b - a} x + \frac{f(a)b - f(b)a}{b - a}$$

whose graph passes through points $(a, f(a))$ and $(b, f(b))$. This procedure gives us the (crude) approximation

$$\int_a^b f(x)\ dx \approx \int_a^b g(x)\ dx = (f(a) + f(b))\ \frac{b-a}{2}.$$

We now suppose that $n > 1$ is an integer, and let $\{x_0, x_1, x_2, \ldots, x_n\}$ be the sequence of $(n+1)$ points defined by

$$x_j = a + \frac{b-a}{n} \cdot j.$$

If we decompose the integral (1) as

$$\int_a^b f(x)\ dx = \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x)\ dx \tag{2}$$

and use the preceding approximation to estimate each of the integrals in (2), then we obtain the estimate

$$\begin{aligned} \int_a^b f(x)\ dx &= \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x)\ dx \\ &\approx \sum_{j=0}^{n-1} (f(x_j) + f(x_{j+1}))\ \frac{b-a}{2n} \\ &= (f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-1}) + f(x_n)) \cdot \frac{b-a}{2n}. \end{aligned} \tag{3}$$

We call (3) the $(n+1)$-point trapezoidal quadrature rule on the interval $[a, b]$. The points $x_0, x_1, \ldots, x_n$ are called the nodes of the $(n+1)$-point trapezoidal rule on $[a, b]$, and the quantities $w_0, w_1, \ldots, w_n$ defined by

$$w_j = \frac{b-a}{2n} \lambda_j,$$

where

$$\lambda_j = \begin{cases} 1 & \text{if } \ j = 0, n \\ 2 & \text{otherwise,} \end{cases}$$

are called the weights of the $(n+1)$-point trapezoidal rule. Note that using this notation, (3) can be rewritten as

$$\int_a^b f(x)\ dx \approx \sum_{j=0}^{n} f(x_j) w_j. \tag{4}$$

Project one consists of writing a function called "traprule" which calculates the $(n+1)$-point trapezoidal rule. More specifically, your routine will take as input an integer $n > 0$ and double precision numbers $a < b$. It will return the nodes $x_0, x_1, \ldots, x_n$ and weights $w_0, w_1, \ldots, w_n$ of the $(n+1)$-point trapezoidal routine in user-supplied arrays. The file "traprule.c" gives the calling syntax for the "traprule" function. Your task is to implement the function as described there.

2. Testing and grading

A public test code is given in the file "traptest1.c". Another test code, called "traptest2.c", will be used to test your function as well. Half of the project grade will come from the first test file, and the second half will come from the second.

The commands

```
gcc -o traptest1 traprule.c traptest1.c -lm
./traptest1
```

can be used to compiler and execute your program. The first command compiles your program and creates an executable file named "traptest1". The "-lm" option tells the compiler to include the math library which implements cosine, sine and other basic functions. The second command runs the traptest1 executable. There are five tests of your function in traptest1, and the program will tell your score out of 5. We will also test your code by compiling against "traptest2.c", which we will not release until after the projects are due.

More explicitly, we will grade your project by running the sequence of command

```
gcc -o traptest1 traprule.c traptest1.c -lm
./traptest1
gcc -o traptest1 traprule.c traptest2.c -lm
./traptest2
```

You will get a 0 on your project if it does not compile and run. Please start work on your project early and come see either myself or our TA, Karry Wong, if you are having difficulties getting it to compile.

Submitting your project:

You will submit your project using canvas. You should submit only your "traprule.c" file. You must submit your file by 11:59 PM on the due data. **Late assignments will not be accepted.**