

1. Project description

In this project, we will write C functions to calculate the Fourier coefficients of a function and to evaluate a function given its Fourier coefficients.

The first of these functions will be called “fourcoefs” and it will, given the capability of evaluating a function f at any point on the interval $[-\pi, \pi]$ and an integer $N > 0$, use the $(2N + 1)$ -point periodic trapezoidal rule to approximate the Fourier coefficients

$$a_{-N}, a_{-N+1}, \dots, a_{-1}, a_0, a_1, \dots, a_{N-1}, a_N$$

of f . These are defined by the formula

$$a_m = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t) \exp(-imt) dt.$$

The $(2N + 1)$ -point periodic trapezoidal rule is

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} g(t) dt \approx \frac{1}{2N+1} \sum_{j=0}^{2N} g\left(-\pi + \frac{2\pi}{2N+1}j\right),$$

so that the desired approximations

$$\widetilde{a_{-N}}, \widetilde{a_{-N+1}}, \dots, \widetilde{a_{-1}}, \widetilde{a_0}, \widetilde{a_1}, \dots, \widetilde{a_{N-1}}, \widetilde{a_N}$$

of the Fourier coefficients of f are given by

$$\begin{aligned} \widetilde{a_m} &= \frac{1}{2N+1} \sum_{j=0}^{2N} f\left(-\pi + \frac{2\pi}{2N+1}j\right) \exp\left(-im\left(-\pi + \frac{2\pi}{2N+1}j\right)\right) \\ &= \frac{1}{2N+1} \sum_{j=0}^{2N} f\left(-\pi + \frac{2\pi}{2N+1}j\right) \exp(im\pi) \exp\left(-im\left(\frac{2\pi}{2N+1}j\right)\right) \\ &= \frac{(-1)^m}{2N+1} \sum_{j=0}^{2N} f\left(-\pi + \frac{2\pi}{2N+1}j\right) \exp\left(-\frac{2\pi i}{2N+1}mj\right) \end{aligned}$$

You can use your code from Project 1 to construct the nodes and weights of the trapezoidal rule, but be aware that the $(2N + 1)$ -point periodic trapezoidal rule on the interval $[-\pi, \pi]$ coincides with the $(2N + 2)$ -point nonperiodic trapezoidal rule on the interval $[-\pi, \pi]$.

The second function you will write will be called “foureval” and it will, given the approximations (1) of the Fourier coefficients computed by the “fourcoefs” functions, evaluate the Fourier series

$$\sum_{n=-N}^N \tilde{a}_n \exp(int)$$

at a specified point t .

The file “fourex.c” gives the calling syntax for the “fourcoefs” and “foureval” functions. Your task is to implement the functions as described there.

2. Testing and grading

A public test code is given in the file “fourtest1.c”. Another test code, called “fourtest2.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 fourtest1 fourex.c fourtest1.c -lm
./fourtest1
```

can be used to compile and execute your program. The first command compiles your program and creates an executable file named “fourtest1”. The “-lm” option tells the compiler to include the math library which implements cosine, sine and other basic functions. The second command runs the fourtest1 executable. There are five tests of your function in fourtest1, and the program will tell your score out of 5. We will also test your code by compiling against “fourtest2.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 fourtest1 fourex.c fourtest1.c -lm
./fourtest1
gcc -o fourtest2 fourex.c fourtest2.c -lm
./fourtest2
```

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 “fourex.c” file. You must submit your file by 11:59 PM on the due date. **Late assignments will not be accepted.**