Experiment-6
Numerical Integration (Simpson's Rule)

(Duration: 105 mins)

Author: Oğulcan Erdoğan
ogulcanerdogan@iyte.edu.tr

Purpose: This lab aims to numerically calculate the integral of a given function using Simpson's rule and its modified version that selects the number of bins adaptively.

Introduction

A considerable improvement over the trapezoidal rule (from the previous week) can be achieved by approximating the function within each of the n intervals by some polynomials. When second-order polynomials are used, the resulting algorithm is so-called *Simpson's Rule*.

If we define h as the width of each of the subdivisions on the x axis and apply Simpson's rule, then the area under the curve in the interval (x, x+h) is approximated by

$$\frac{h}{6}\left[f(x) + 4f\left(x + \frac{h}{2}\right) + f(x+h)\right] \tag{1}$$

For more information, see your course notes in which you can find some examples of how to implement this rule and its modified versions.

Problem Statement

In Figure 1, you can see a waveform that is well-known by the communication engineers and produced for this experiment (it could be different in practice).

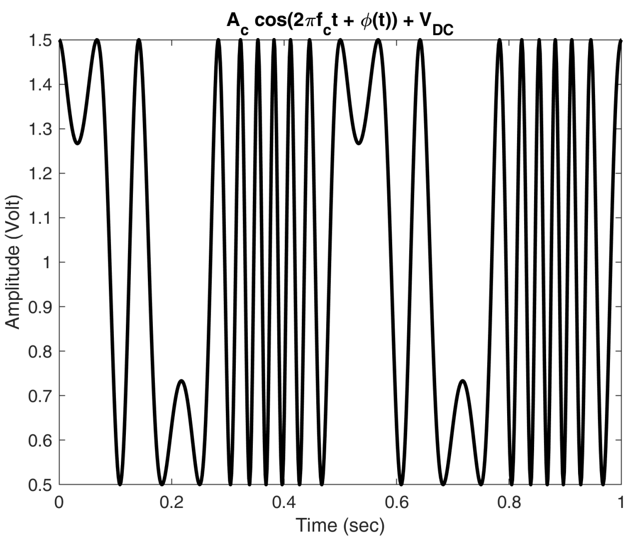

Figure 1: Frequency Modulation (FM) Signal

As you can see from this figure, sometimes the function changes rapidly, and sometimes it changes slowly. Therefore, the integral calculation may diverge if you do not track this function properly. Fortunately, you will see such a nice method to overcome this difficulty.

Let the waveform is defined as

$$x(t) = 0.5\cos(20\pi t + \frac{d}{dt}\phi(t)) + V_{DC} \tag{2}$$

where $V_{DC} = 1V$ is the DC voltage to give an offset and $\phi(t)$ is the phase that is changing with the time defined as

$$\phi(t) = \sin(4\pi t) \tag{3}$$

Lab Procedure

After the necessary definitions of these functions, you should follow the below prototypes and steps:

```
float phi_t(float t){
float phi_diff(float t, float h){
float xt(float t, float h){
float simpson(float (*f)(float, float), int n){
float adaptive_simpson(float (*f)(float, float), int n0, float tol){
void main(){
```

Figure 2: Given Prototypes

1. Define start and end points of the interval $t \in [0, 10]$, and π as 3.142857 (if necessary) with define statements (be careful about integer division). (**2pts**)
2. Write a function for the $\phi(t)$ given in Eq.(3). (**8pts**)
3. Write a function that calculates the numerical derivative of the $\phi(t)$ can be evaluated by using below numerical differentiation formula: (**10pts**)

$$\frac{d}{dt}\phi(t) = \frac{\phi(t+h) - \phi(t)}{h} \tag{4}$$

4. Write a function for the $x(t)$ given in Eq.(2). (**15pts**)
5. Write a function for Simpson's rule to calculate the integral of $x(t)$ with given interval. (**25pts**)
6. Write another function that calculates the integral of $x(t)$ by Simpson's rule with adaptive bins. (**30pts**)
7. Choose the tolerance value for the adaptive bin selection as 10^{-4} (if this slow down your computer choose 10^{-3} or 10^{-2}). Print both the Simpson's and the Adaptive Simpson's results as in Figure 3 for each number of panels n which goes from 10 to 200. (**10pts**)

Using *printf()* elsewhere from *main()* is strictly prohibited!

Number of Panels	Simpson's	Adaptive Simpson's
10	14.943645	10.002582
20	8.347838	10.002582
30	8.866662	10.002625
40	13.319594	10.002582
50	7.666779	10.002712
60	12.427437	10.002625
70	10.158974	10.002565
80	11.246348	10.002582
90	9.977753	10.002843
100	10.780578	10.002712
110	8.787088	10.002584
120	10.212627	10.002625
130	9.763902	10.002562
140	10.461587	10.002565
150	10.071018	10.003041
160	8.473171	10.002582
170	10.119150	10.002926
180	10.707200	10.002843
190	9.850471	10.002925
200	9.798398	10.002712

Figure 3: Screenshot of the output