APPM 4600 Lab 12
Adaptive Quadrature

1 Overview

In this lab, you will investigate the performance of adaptive quadrature and how that relates the underlying quadrature that is being applied to each sub-interval.

2 Before lab

1. Create subroutines for evaluating :
   - Composite trap on an interval (input: $a, b, f(x)$ and N number of points) Composite Simpsons rule on an interval (input: $a, b, f(x)$ and N number of points)
2. Watch the video below about adaptive quadrature

3 Lab Day

The code named `adaptive_quad.py` implements the adaptive quadrature as described in the pre-lab video. The underlying quadrature that is used in each sub-interval is Gaussian quadrature.

Go through the code and examples with the TA. Discuss the results.

4 Exercises

1. Create three adaptive quadrature codes using Composite Trapezoidal, Composite Simpsons and Gaussian quadrature. (Note: you can build this directly from the provided adaptive Gaussian quadrature code by changing one line and giving the resulting subroutine a different name.)
2. Let $n = 5$ denote the number of nodes used on each sub interval. Use each of the quadrature schemes to approximate
$$\int_{0.1}^{2} \sin(\frac{1}{x})dx$$
to within 10^{-3}. How many intervals does each method need to get the desired accuracy? Which is the the best choice of adaptive scheme?

4.1 Deliverables

All codes should be pushed to git and your responses to the questions should be entered into Canvas.