NUMERICAL ANALYSIS

MAT 360

# Homework 04/23

Break the adaptive integration code:

## What makes it fail?

## Asymptotes:

The code fails when the area we want to integrate has a vertical asymptote inside. It fails in two different ways:

* If nailing the asymptote: When the choices of a and b leads the new value of c to nail the asymptote, when the program calculates the value of f(c) it find the value ±∞ and the calculation will fail. An example of this behavior is evaluating the function f(x) = 1/x on a symmetric interval from x = 0, like a = -1 and b = 1. When executed on this conditions, the program will return:

* If not nailing but trying to calculate area next to asymptote: If the function has an asymptote and we choose the interval in a way that the program will not nail the asymptote, but it is included inside, the program will fail and calculate a wrong output. We can observe this when we evaluate f(x) = 1/x on the interval a = -1 and b = 2. The output will be completely different to the Mathematica’s result (which is failing too) and it will be wrong. For this case it will return , but we know that for this function the result will be the same as for evaluating the function on a = 1, b = 2 due to the symmetry cancelation. For this values, the correct output is , agreeing with Mathematica’s result.
  1. Non defined interval for the function:

When the function is evaluated in an interval where the function is not defined, the program will obviously fail. An example is evaluating f(x) = ln(x) on the interval a = -3 and b = -2.

* 1. Value of a > b:

If the value of the start point of the interval is bigger than the endpoint, the program will fail.

* 1. Too wild function:

If we try to evaluate a function that becomes too wild and so it needs too many subintervals for the precision required. Then the function will stop iterating and the final value may have a big error.

## How can you improve it?

## Asymptotes:

The code fails when the area we want to integrate has a vertical asymptote inside. It fails in two different ways:

* If nailing the asymptote: Improve the program making it recognize that if f(c) is equal to ±∞ it returns an error message.

* If not nailing but trying to calculate area next to asymptote: Improve the program making it recognize that if the c in each new sub interval grows wildly approximating f(c) to ±∞, it will return an error message due to a possible asymptote.
  1. Non defined interval for the function:

Improve the code to evaluate the function on the interval to check that it is defined in it.

* 1. Value of a > b:

Create a piece of code that checks that a is less than b and if not, switches their values.

* 1. Too wild function:

This is actually not a fail of the program, because the iterations constraint is taking care of stop the program on purpose when the function becomes too wild. If you want, you can make the number of iterations bigger, or the precision smaller, but there is always going to be a machine precision constraint.