Victoria (Davis) Wegener

Dr. Long

MAT 360

23 April 2015

Adaptive Quadrature Testing

 The adaptive quadrature code may seem easy to break at first, but after testing it a little bit, it is actually quite difficult. From trying easy functions, like f[x] = x and f[x] = Sin[x], to trying more difficult functions, like f[x] = -e^(x^2) and f[x] = Log[Log[x]], the code would not fail. Improving the code, after making it fail, is even more difficult.

 The one of the first, more difficult, functions I tried was f[x] = Sin[x] + Cos[x]. Which I thought would break it, but the code seemed to have no problem with the subdivision and graphing process. Then, I knew that this was not going to be an easy code to break and that I would need a much more difficult function to get this code to fail.

 Knowing I needed more pressure, I started thinking about Logs. So I tried f[x] = Log[x], which of course, the code did with much ease. But, could it do f[x] = Log[Log[x]]? The code seemed to have some difficulty with this function. Finally, I had a lead. The only difficulty was with the code producing negative infinity. Thus, I was not satisfied with this, since this technically may not be the code failing, but maybe the interval. So, I continued my search for the function that would make it fail.

 Well, if it had difficulty with f[x] = Log[Log[x]], would it have difficulty or even fail with f[x] = Log[Log[x]]^2. Although, the code seemed to have difficulty with this function it seemed to be the same issue as when f[x] = Log[Log[x]]. The code was just giving a positive infinity. So, I was again not satisfied with this function making the code fail, it just seemed to be an error that the adaptive quadrature code was not making, but a human error.

 Continuing on the long trek, I decided to try some easier functions. I started with f[x] = 1/x. Oh, and I was so excited to see the code fail with this. A huge sigh of relief, the code finally failed. As I read the error just to double check my work, I read that is was impossible because you cannot divide by 0. So, I knew I had just been tricked by the code. The code did not fail. The interval had failed on me. The trek continued.

 Finally, I decided to go back to cosines and sines. Instead of adding them, what would happen if I multiplied them. So, I tried f[x] = (Cos[x]^3)(Sin[x]^2). Of course, this would not make the code fail either. Why not try f[x] = (Cos[x]^2)(Sin[x]^3). I mean, I have nothing to lose at this point. I was tired and the code was making me fail. But then, a miracle happened. The code seemed to not like f[x] =(Cos[x]^2)(Sin[x]^3). I was highly suspicious. I was not going to be tricked again! But this time, the code would not graph it three times the same. The graphs did not match! So, as far as my analysis goes f[x]= (Cos[x]^2)(Sin[x]^3) made the code fail.

 In conclusion, f[x] = (Cos[x]^2)(Sin[x]^3), seemed to make the code fail. The subdivision graph was a straight line. Whereas, we know that f[x] = (Cos[x]^2)(Sin[x]^3) is not a line. Unfortunately, I don’t know how to improve the code. Especially since f[x] = (Cos[x]^3)(Sin[x]^2) did not make the code fail. I tried messing with the interval, eps, and its, but nothing changed the outcome.