CS 4250 / MATH 4220 **Homework 4** **Due Oct 13th, 2016**
Alex Townsend

You are encouraged to discuss the problems with your peers. You must hand in individual code.

1. There is only one question this week. A company has asked you to calculate 30 integrals. The task is to write a command `<netid>_myquad( k )` that numerically computes the integral

$$\int_a^b f_k(x)dx$$

and returns its value (see below for $f_k(x)$). The aim is a `<netid>_myquad( k )` command that delivers a fast and accurate value for the integral of the kth function (an "accurate" answer is function-dependent as it depends on the condition number of the problem, see hw2 Q1). If your command accurately computes the integral of the kth function, then you will receive 1 point. If your `<netid>_myquad( k )` command is faster than 0.01s (on our machine[1] when we run your code) at calculating the kth integral, then you will receive 1 point. Therefore, there are 2 possible points per integral for a maximum number of 60 points. A full score means your command computes every integral fast and accurately. A skeleton MATLAB script is provided online and is called `ajt253_myquad( k )`.

Some integrals may be ill-conditioned problems or divergent, some may have integrands that have cancellation errors, some are smooth for Gaussian quadrature, some are designed for spline or piecewise polynomial interpolation, and others are noisy for least squares fits. Looking at each one you should be able to design a good algorithm. If an integrand has cancellation errors, then you should fix it first to receive the accuracy point for that integral. For an integrand with noise we want an accurate expected value of the integral.

There is a design decision here to maximum your score. You may give up on integral k because it requires too many hours of your time to integrate accurately, or you may calculate integral k using a suboptimal algorithm to avoid implementing another method, or you may focus on accuracy but not speed. For example, the command that returns `NaN` on every integral will receive 30/60 because performing no computation will result in code that runs in less than 0.01 seconds.

Here are the 30 functions to integrate with a corresponding interval to integrate over. Please download `ajt253_myquad.m` from the course webpage to get started, and **make sure you rename the command** `<net_id>_myquad`.

Please hand in two things: (1) A PDF briefly explaining the choices you made, the cancellation errors you avoided, and the integrals, if any, that are impossible (explaining why). For example, if an integral is divergent and you explain why in writing, then you will receive 1 point for your accuracy point for that integral. (2) Your `<net_id>_myquad.m` command. Upload to GitHub, see instructions below.

[1]Our machine is very likely going to faster than your laptop.

$$
\begin{aligned}
f_1(x) &= e^x && \text{on } [0,1]\\
f_2(x) &= \begin{cases} 1, & x \geq 0.3,\\ 0, & x < 0.3, \end{cases} && \text{on } [0,1]\\
f_3(x) &= x\sin(1/x) && \text{on } [0,1]\\
f_4(x) &= \log(1+x) && \text{on } [0,10^{-12}]\\
f_5(x) &= \frac{1}{\cosh(20(x-.2))} + \frac{1}{\cosh(400(x-.4))} + \frac{1}{\cosh(8000(x-.6))} && \text{on } [0,1]\\
f_6(x) &= \frac{1}{x^2+1.005} && \text{on } [-1,1]\\
f_7(x) &= \cos(x) + 10^{-2}\epsilon, \quad \epsilon \sim \mathcal{N}(0,1) && \text{on } [-1,1]\\
f_8(x) &= 25e^{-25x} && \text{on } [0,10]\\
f_9(x) &= \frac{e^{10x}-1}{e^{(101/10)x}} && \text{on } [0,100]\\
f_{10}(x) &= x^2+x+1 && \text{on } [0,1]\\
f_{11}(x) &= \lfloor e^x \rfloor && \text{on } [0,3]\\
f_{12}(x) &= \sqrt{x^3} && \text{on } [0,1]\\
f_{13}(x) &= x^{-1/2} && \text{on } [0,1]\\
f_{14}(x) &= (x<1)(x+1) + (1 \leq x \leq 3)(3-x) + 2(x>3), && \text{on } [0,5]\\
f_{15}(x) &= \frac{1}{1+x^4}, && \text{on } [0,1]\\
f_{16}(x) &= \sin(100\pi x)/(\pi x) && \text{on } [0,1]\\
f_{17}(x) &= \log(x) && \text{on } [0,1]\\
f_{18}(x) &= T_{10000}(x) = \cos(10000\cos^{-1}(x)) && \text{on } [-1,1]\\
f_{19}(x) &= T_{100}(x) = \cos(100\cos^{-1}(x)) && \text{on } [-1,1]\\
f_{20}(x) &= \det(H - xI_{50}), H_{jk} = 1/(j+k-1), 1 \leq j,k \leq 50 && \text{on } [-1,1]\\
f_{21}(x) &= (x \leq 0)\sin(\pi x) + (x \geq 0)\cos(\pi x) && \text{on } [-1,1]\\
f_{22}(x) &= \frac{\cos(\pi x/2)}{10^9} && \text{on } [-1,1]\\
f_{23}(x) &= \exp(-x) && \text{on } [0,\infty]\\
f_{24}(x) &= \frac{\log(2x)}{\log(x)} && \text{on } [0,1/2]\\
f_{25}(x) &= (-1)^n/n \text{ for } (n^{-1} <= x < (n+1)^{-1}) && \text{on } [0,1]\\
f_{26}(x) &= \sin(5\sin x) && \text{on } [0,\pi]\\
f_{27}(x) &= e^{x^2} && \text{on } [-1,1]\\
f_{28}(x) &= \sin(x)/\ln(x) && \text{on } [0,1]\\
f_{29}(x) &= 1/\sqrt{1-x^2} && \text{on } [-1/2,1/2]\\
f_{30}(x) &= \ln(2x) - \ln(2) && \text{on } [0,1]
\end{aligned}
$$

To calculate your score we will be running your script on our server so that we can calculate timings. Once you are done, please upload your file at `https://github.com/ajt60gaibb/MATH4250` before 11:40am on Thursday the 13th of October (if you upload beforehand, then you can make changes until 11:40am). To upload the file on `https://github.com/ajt60gaibb/MATH4250` press the button "Upload files".