MA-UY 4423 Worksheet 1
In class, March 31, 2015

The following are problems to be worked on during class-time, with or without the collaboration of fellow students.

There is no grade for these problems, the goal is to reinforce your understanding of the material and get some programming practice in.

**Problem 1:** A Gaussian Quadrature Rule

The first two Legendre polynomials are:

\[ P_0(x) = 1, \quad P_1(x) = x, \]

and are defined for \( x \in [-1, 1] \). Subsequent Legendre polynomials can be calculated using the recurrence relation:

\[ P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x). \]

They are orthogonal with respect to the weight \( w(x) = 1 \) on the interval \([-1, 1]\):

\[ \int_{-1}^{1} P_m(x) P_n(x) \, dx = \frac{2}{2n+1} \delta_{mn}. \]

The usual \( k \)-point Gaussian quadrature associated with the Legendre polynomials has as its nodes \( \{x_j\} \) the roots of \( P_k(x) \) on \([-1, 1]\), and the weights satisfy:

\[ w_j = \frac{2}{k P_{k-1}(x_j) P'_k(x_j)}. \]

The derivative of \( P_n \) can be calculated as:

\[ P'_0(x) = 0 \]
\[ P'_1(x) = 1 \]
\[ P'_{n+1}(x) = \frac{2n+1}{n+1} \left( x P'_n(x) + P_n(x) \right) - \frac{n}{n+1} P'_{n-1}(x). \]

1. Write a subroutine that calculates a \( k \)-point Gaussian quadrature by first finding the roots of \( P_k \) and then calculating the weights.

2. How do you know that your nodes and weights are correct? How accurate are they?
Problem 2: Adaptive Gaussian Quadrature

Adaptive quadrature is commonly used as a one-stop shopping solution to solve many problems in numerical integration. For example, let our quadrature rule be a $k$-point Gaussian rule (that you are able to compute using the solution to Problem 1). For a given error tolerance $\varepsilon$, adaptive quadrature proceeds as follows:

Step 1: Initialize.
$$\int_{-1}^{1} f(x) \, dx \approx \sum_{j=1}^{k} w_j f(x_j) = I_0$$

Step 2: Subdivide.
$$\int_{-1}^{0} f(x) \, dx \approx \sum_{j=1}^{k} \frac{w_j}{2} f\left(\frac{x_j - 1}{2}\right) = I_{10},$$
$$\int_{0}^{1} f(x) \, dx \approx \sum_{j=1}^{k} \frac{w_j}{2} f\left(\frac{x_j + 1}{2}\right) = I_{11}.$$ 

Step: Test.

If $|I_{10} + I_{11} - I_0| < \varepsilon$ then stop, and declare that the integral be approximated by $I_{10} + I_{11}$. If not, repeat recursively on the smaller intervals $[-1,0]$ and $[0,1]$.

1. Write a subroutine which receives as input an integer $k$, an interval $a,b$, and precision $\varepsilon$, and calculates the integral of a function $f$ on that interval using adaptive Gaussian quadrature. Your subroutine should also return the number of intervals that were required to achieve the precision $\varepsilon$.

2. Test your routine for various values of $k$ on the integral:
$$\int_{-5}^{5} \frac{dx}{x^2 + 1} = \arctan(5) - \arctan(-5)$$

Problem 3: Trapezoidal rule for $J_0$

The Bessel function of the first-kind of order zero $J_0$ is the solution to the following ODE:

$$x^2 y''(x) + x y'(x) + x^2 y(x) = 0,$$
$$y(0) = 1,$$
$$y'(0) = 0.$$
It can also be calculated using the integral formula:

\[ J_0(x) = \frac{1}{\pi} \int_0^\pi \cos(x \sin(\theta)) \, d\theta. \]

This is a periodic integral, and can be efficiently evaluated using the trapezoidal rule. Make a table of varying \( x \) from 1 . . . 1000 along with the number of quadrature nodes needed to obtain \( J_0(x) \) to an absolute precision of \( 10^{-10} \). How can you estimate the number of nodes needed for a given \( x \) based on the information in this table?