APPM 4600 — HOMEWORK # 6

For all homeworks, you should use Python. **Do not use** symbolic software such as Maple or Mathematica.

1. Find the constants x_0, x_1 and c_1 so that the quadrature formula

$$\int_0^1 f(x)dx = \frac{1}{2}f(x_0) + c_1 f(x_1)$$

   has the highest possible degree of precision.

2. Consider the definite integral $\int_{-5}^{5} \frac{1}{1+s^2} ds$.

   (a) Write a code to approximate it using a composite Trapezoidal rule. To do this, partition the interval $[-5, 5]$ into equally spaced points $t_0, t_1, \ldots, t_n$.
   Write another code to approximate $\int_{-5}^{5} \frac{1}{1+s^2} ds$ using a composite Simpson's rule. To do this, partition the interval $[-5, 5]$ into equally spaced points $t_0, t_1, \ldots, t_n$ where $n = 2k$ is even. The even indexed points should be the endpoints of your subintervals.
   You may combine the two into one code that selects the desired method if you wish.

   b) Use the error estimates derived in class to choose n so that

$$\left| \int_{-5}^{5} \frac{1}{1+s^2} ds - T_n \right| < 10^{-4} \quad \text{and} \quad \left| \int_{-5}^{5} \frac{1}{1+s^2} ds - S_n \right| < 10^{-4},$$

   where T_n is the result of the composite Trapezoidal rule and where S_n is the result of the composite Simpson's rule. Compute the actual error for both and compare with the target 10^{-4}.

   c) Run your code with the predicted values of n and compare your computed values S_n and T_n with that of `SCIPY`'s `quad` routine on the same problem. Run the built in quadrature twice, once with the default tolerance of 10^{-6} and another time with the set tolerance of 10^{-4}. Report the number of function evaluations required in both cases and compare these to the number of function values your codes (both S_n and T_n) required to meet the tolerance

3. Assume the error in an integration formula has the asymptotic expansion

$$I - I_n = \frac{C_1}{n\sqrt{n}} + \frac{C_2}{n^2} + \frac{C_3}{n^2\sqrt{n}} + \frac{C_4}{n^3} + \cdots$$

   Generalize the Richardson extrapolation process to obtain an estimate of I with an error of order $\frac{1}{n^2\sqrt{n}}$. Assume that three values I_n, $I_{n/2}$ and $I_{n/4}$ have been computed.

4. The gamma function is defined by the formula

$$\Gamma(t) = \int_0^\infty x^{t-1}e^{-x}dx, \quad t > 0.$$

   Write a program to compute the value of this function from the definition using each of the following approaches:

(a) Truncate the infinite interval of integration and write a composite trapezoidal rule code to perform the numerical integration. You will need to do some experimentation or analysis to determine where to truncate the interval based upon the usual trade-offs between accuracy and efficiency. Please describe your reasoning for your choice of interval and step size for the Trapezoidal Rule. Compare the relative accuracy of this solution with the value given by the Python gamma function (`scipy.special.gamma`) at $t = 2, 4, 6, 8, 10$ (Recall $\Gamma(k) = (k-1)!$ for positive integer k) .

(b) Use the Matlab adaptive quadrature routine `quad` to solve the above integral on the same interval you used for part (a). Compare the accuracy of this solution with the one you obtained in part (a) at the same values of t. Also, compare the number of function evaluations required by the two methods.

(c) Gauss-Laguerre quadrature is designed for the interval $[0, \infty)$ and the weight function $w(x) = e^{-x}$. It is therefore ideal to use for this problem. Call the Numpy subroutine `numpy.polynomial.laguerre.laggauss` to to obtain the n weights $\mathbf{w}$ and n abscissae $\mathbf{x}$ for Gauss-Laguerre quadrature and use this to approximate $\Gamma(t)$. Keep in mind Gauss-Laguerre is a generalized gaussian rule of the form:

$$\int_0^\infty f(x)e^{-x}dx \approx \sum_{i=0}^{n} w_i f(x_i)$$