PHYS3437: Computational Methods in Physics, Assignment 4

As well as the requested answers for each question, please email me a copy of your code.

Q1. The *mid-point rule* for integration has a local truncation error proportional to h^3 despite using only one function evaluation (the global error for the composite rule is proportional to h^2). The formula for the mid-point rule is

$$\int_b^a f(x)dx = (b-a)f\left(\frac{a+b}{2}\right) + \frac{(b-a)^3}{24}f''(a) + ... \simeq hf(x_{mid}) + \mathcal{O}(h^3), \tag{1}$$

and we associate h with the interval width $b-a$. When applied as a composite rule in a set interval $[a,b]$, where the width of the n sub-intervals are given by $h=(b-a)/n$, the remainder term for this composite Mid-point Rule is given via,

$$\int_b^a f(x)dx \simeq h\sum_{i=1}^{n} f(a+(i-1/2)h) + \mathcal{O}(h^2), \tag{2}$$

i.e. for the composite rule the error is proportional to h^2.

Write a code that uses a composite version of the mid-point rule with n=1,2,4,8,16 sub-intervals to evaluate

$$\int_0^1 e^x dx = e-1$$

Suppose we define n using n=2^{k-1}, then n=1,2,4,8,16 correspond to k=1,2,3,4,5. If we define the estimate of the above integral for a given value of k to be I_k (*i.e.* I_3 corresponds to the n=4 estimate) then for a given pair I_k and I_{k-1} Richardson Extrapolation allows us to define a solution accurate to the next highest order in h^2, which we define as $I_{k,2}$ by

$$I_{k,2} = I_k + \frac{I_k - I_{k-1}}{3}.$$

In this case, solutions accurate to h^2 are used to give an answer accurate to h^4. In general, solutions accurate to h^{2j} can be calculated by

$$I_{k,j} = I_{k,j-1} + \frac{I_{k,j-1} - I_{k-1,j-1}}{4^{j-1}-1}. \tag{3}$$

where $I_k \equiv I_{k,1}$. Extend your code to use the values of I_k you calculated earlier to evaluate the Romberg Table of $I_{k,j}$ values and print the table (once you have the column of $I_k \equiv I_{k,1}$ values, calculating the next column of values in the table is performed using equation (1)) . Contrast the accuracy of the $I_{5,5}$ value to the true solution with that of the highest accuracy I_k solution you previously calculated. **Solutions using a different approach to the evaluation of $I_{5,5}$ are NOT acceptable.** (NOTE the algorithm outlined is not actually quite correct from an error cancellation perspective, but on average it works very well. It has the advantage of not requiring evaluations at the beginning or end of the integration interval.)

Q2. The differential equation

$$y'(t) = \frac{y(1-y)}{2y-1}$$

with $y_0 = 5/6$ has the solution

$$y(t) = \frac{1}{2} + \left[\frac{1}{4} - \frac{5}{36}e^{-t}\right]^{1/2}.$$

(a) Write codes to solve the equation using three of the methods we discussed in the lectures, namely the forward Euler method, Improved Euler method and Runge-Kutta.
(b) Measure the error at step t=2.0 for the different methods and plot (log-log) up the error relative to the exact solution versus the step-size h (consider h values down to 0.0001). What order of convergence do you obtain for each method?

Q3. Sub-random sequences. Generate the first 10,000 points in the triplet series (x_1^i, x_2^i, x_3^i) where the x_j^i are given by Halton sequences for bases 3,5,7. Plot your results on a 3d graph. (HINT: You can always work out the number of digits of a number j in base n by evaluating $\log_n j$. Once you have the total number of digits you need to divide by successive powers of the base to get the digits (think about doing this in base 10, you can divide by powers of 10 and then take the modulus of the resulting number with 10 to get the digit). Once you have the digits you can reverse them and construct the decimal value in the sequence.)

Q4. (a) Write a code to integrate

$$I = \int_0^1 \frac{dx}{1+x^2} = \frac{\pi}{4}$$

using a Monte Carlo method. As a random number generator you can use the `RAN3()` subroutine on the class website. For a 1-dimensional Monte Carlo integral, the standard deviation defines the relative convergence to the mean of f via

$$\int_a^b f dx \simeq < f > \times(b-a) \pm \sqrt{\frac{< f^2 > - < f >^2}{N}} \times (b-a).$$

Calculate the standard deviation and plot it (log-log) as a function of N for $N = 10^2, 10^3, 10^4, 10^5, 10^6$.
(b) Calculate the same integral this time using a sequence of points generated from the Halton's sequence (base 3) you developed in question 3. Plot up the new standard deviation for the same N on the same plot as part (a).
(c) Finally, plot up (log-log) the convergence to the absolute value as a function of N (*i.e.* plot $\pi/4$-your estimate) for the randomly sampled MC estimate in (a) and compare that with the answer from (b). What is the approximate slope for the result of (a), what is the slope for the result of (b). Which method converges faster to the answer and why?