1. Write a subroutine to approximate the solution $y(t)$, to the IVP $y'(t) = f(t, y)$, $y(a) = \alpha$, $t \in [a,b]$, using an order 3-4 multistep predictor-corrector method, but with uniform step size. Call this subroutine mspc34 (in the file mspc34.m).

   \[ \text{function } w = \text{mspcc34}(f,a,b,\alpha,N) \]

   The subroutine should take as input the name of the function $f$, the endpoints $a$ and $b$ of the interval $[a,b]$, an integer $N$ representing the number of subintervals, and the initial value $\alpha$. Your code should return a vector approximation $w$ to the function $y$ over $[a,b]$:

   \[
   (w_1, w_2, \ldots, w_{N+1}) \approx (y(t_0), y(t_1), \ldots, y(t_N)),
   \]

   where $t_j = a + jh$ and $h = (b - a)/N$. Use the A-B 3 step as the predictor and the A-M 3 step as the corrector. This should require only 2 new function evaluations per time step, but will require your rk4 to generate $w_1$ and $w_2$. You need a working rk4 (program assignment 4) to do this; please see me if you need help.

2. Write a main program that tests your subroutines.

   (a) Write a function fnty.m that evaluates $f(t, y) = -20(y - t^2) + 2t$. You should have this already from program 4 assignment.

   (b) Call rk4 and mspc34 with input $a = 0$, $b = 1$, $\alpha = 1/3$ and $N$, for $N = 5, 20, 80, 320, 1280$ and print a table relating $h$, and the errors for each method at $t = 1$ (the solution is $y(t) = t^2 + \frac{1}{3}e^{-20t}$, so the error at $t = 1$ is $1 + e^{-20}/3 - w(N+1)$).