Karin Metzgar
ASTE 546
HW #3

Part 1:

a) B = 1, initialized at 0, a is defined as 1 in a global variable and that is added to 0
   B = 3, at step 2 the a inside the for loop is not relevant outside the for loop, a =2 and that is added to 1
   B = 5, the a defined in my_fun() is not returned to the main function and is therefore discarded, so a is still 2 and it is added to B again.
b) A = 0, initialized as 0 in main and passed into f1, a is 1 in f1 but that value isn't returned to main
   A = 1, we pass the location of a into f2, f2 adds 1 to the pointer to a, so even though the function doesn't return the new value or a it is updated in main?
   A = 2, a is now 1, the location of a is now passed into f3 and one is added, because we are passing in the location of a the function acts like it outputs the value of a.
c) B.x is set to 10 and A.x is set to some arbitrary location/value? They and passed info f1, m1.x = 10 since m2.x = 10 but nothing is passed back into main because pointers aren't used and the function doesn't return the values.
   F1: A.x = ? and B.x = 10
   F2: A.x = ? and B.x = 10, again the value of 10 for A.x / m1.x is not returned to main
   F3: A.x = 10 and B.x = 10 Now that the location of m1 is A.x, it is set to 10 and "returned" to main
   F4: A.x = 10 and B.x = 10
   F5: A.x = 10 and B.x = 10
   F6: will not compile, cannot modify the value or m1.x as its brought in as a constant value ?

Part 2: Finite Difference
Am I missing something?

Derive the 2nd order accurate terms for one-sided forward and backward Finite Difference representations for the 1st derivative $\partial f/\partial x$:

forward: $f(x+\Delta x) = f(x) + \frac{\Delta x}{1!}\frac{\partial f}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f}{\partial x^2} + HOT$

$$\frac{\partial f}{\partial x} = \frac{-f(x)+f(x+\Delta x)}{\Delta x} + \frac{\Delta x}{2}\frac{\partial^2 f}{\partial x^2}$$

$$f'(x) = \frac{f(x+dx)-f(x)}{dx} + \frac{dx}{2}f''(x)$$

backward: $f(x-\Delta x) = f(x) - \frac{\Delta x}{1!}\frac{\partial f}{\partial x} + \frac{(\Delta x)^2}{2!}\frac{\partial^2 f}{\partial x^2} - \ldots + HOT$

$$\frac{\partial f}{\partial x} = \frac{f(x)-f(x-\Delta x)}{\Delta x} + \frac{\Delta x}{2}\frac{\partial^2 f}{\partial x^2}$$

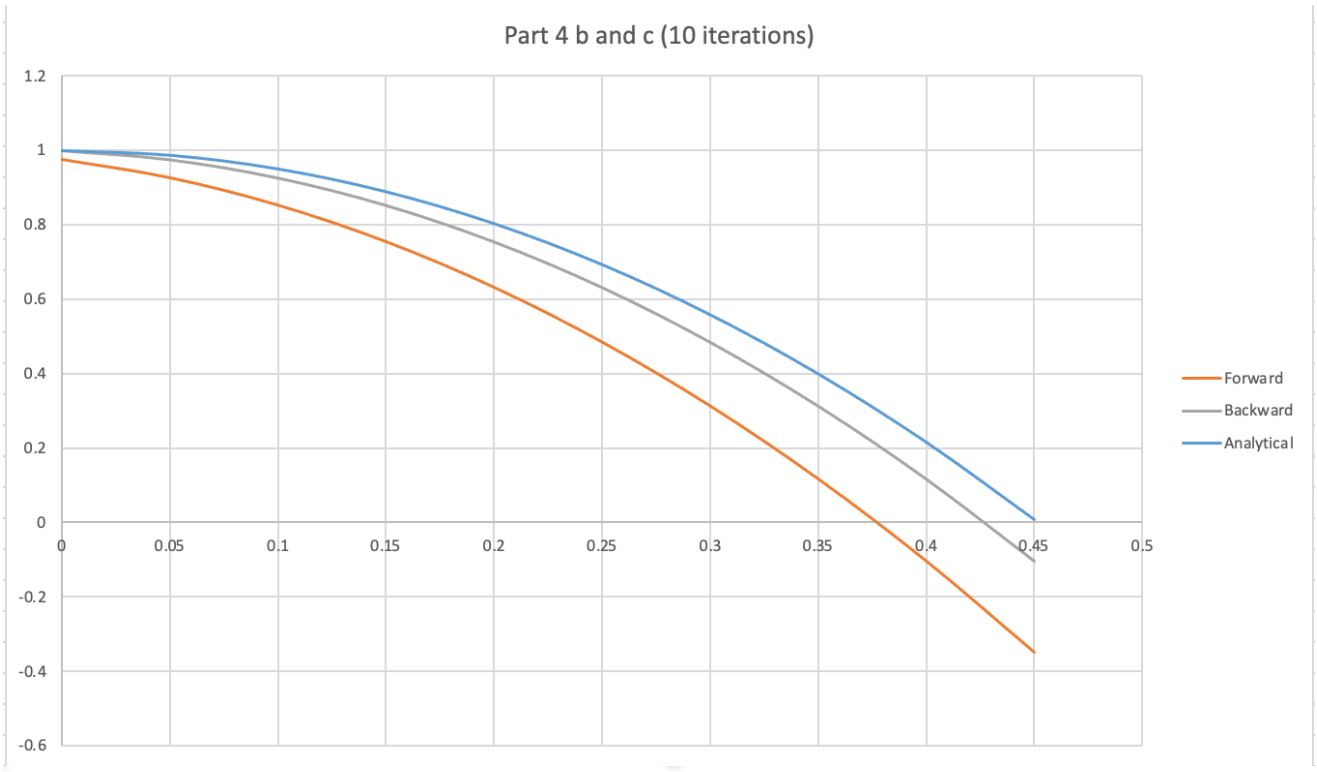
Part 3: Integration

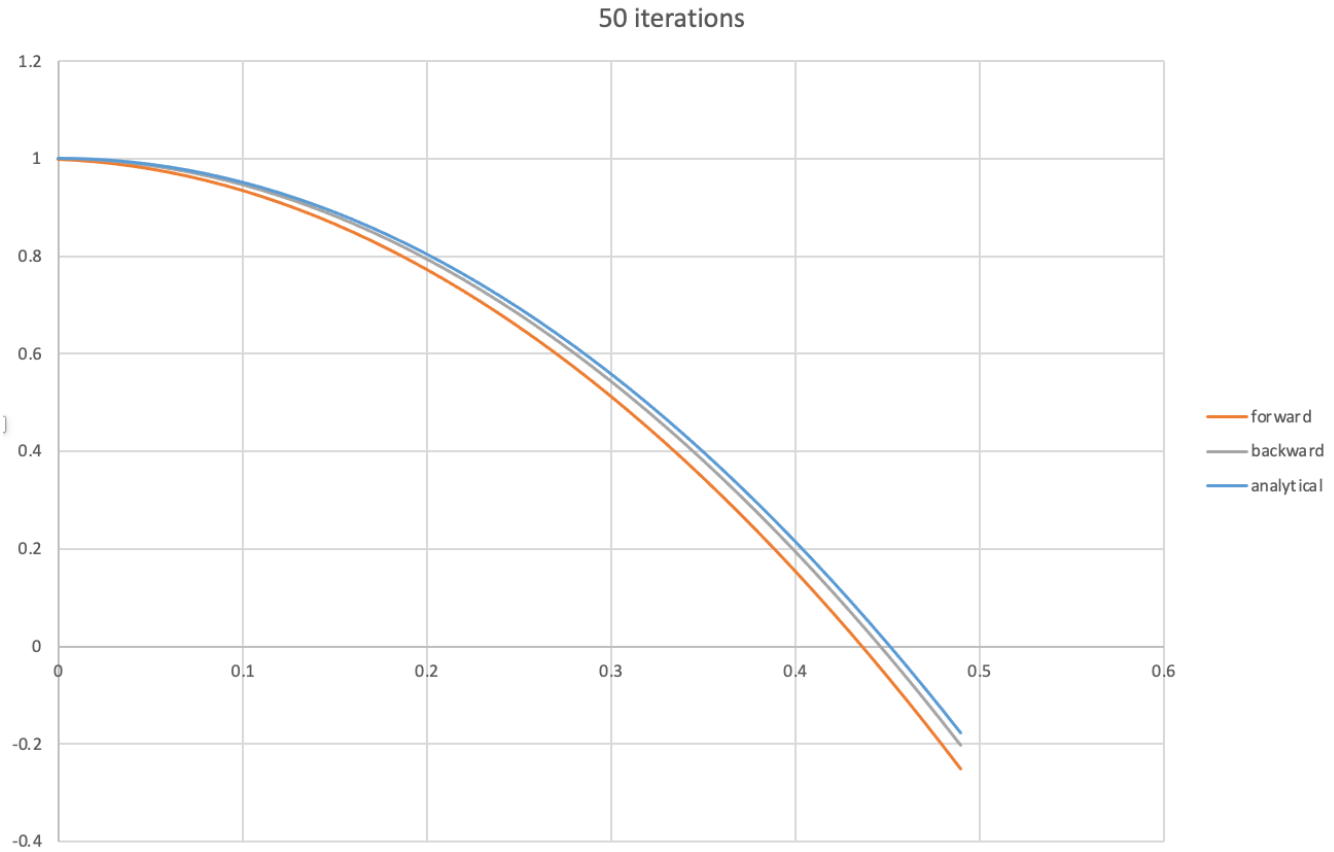
A) $y = y0 - v0(t) + \frac{1}{2}at^2$ or $y = y_0 + \frac{-9.81}{2}t^2$

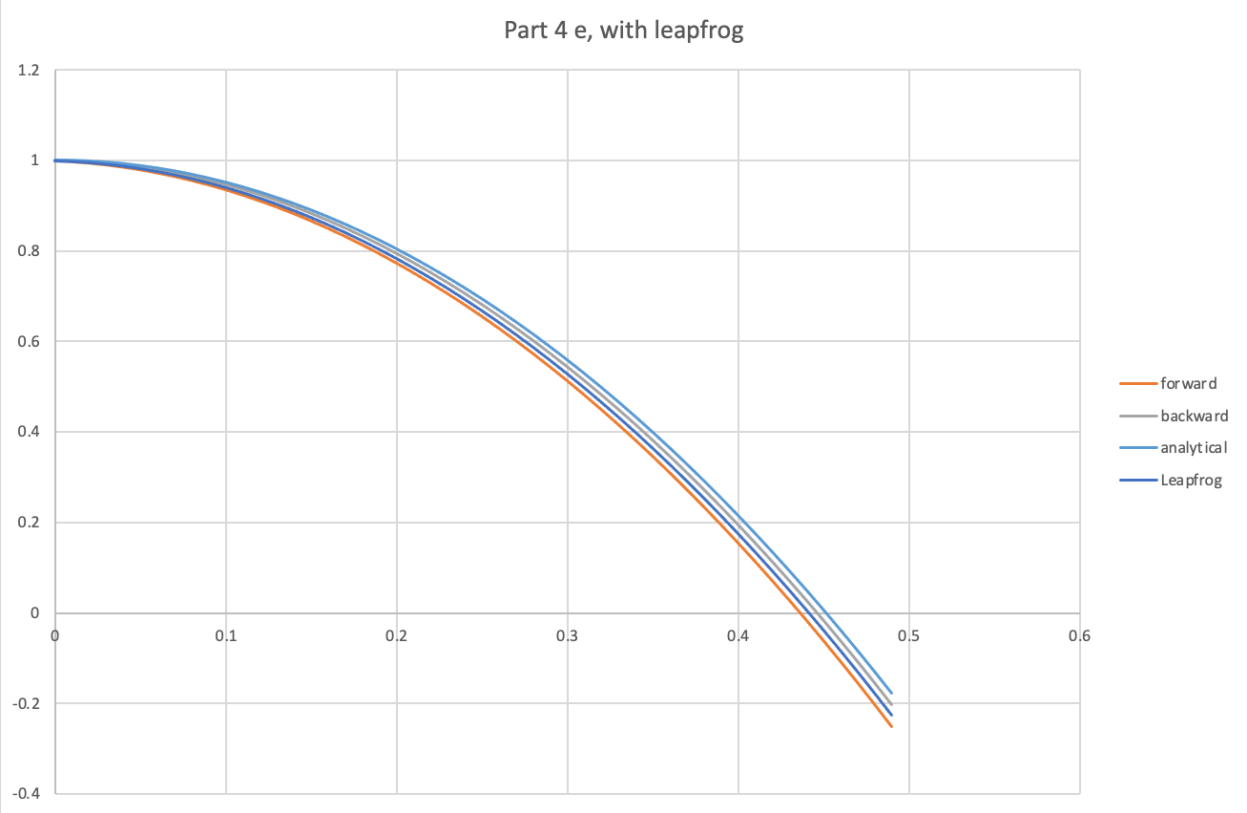
I am surprised that in my data the Analytical solution is not between the forward and backward Eulerian approximations ..

I used the external variable to store the value of the previous iterations velocity and position in all cases, not sure if this is what was recommended but struggled to do it without using an array.

```
for (int i = 0; i < iterations; i++) {
    t = static_cast<double>(i) * dt;

    //calculate velocity
    v = v_old + (g * dt);

    //b
    //calulate y with backward Euler
    y_backwardE = y_old_b + (v * dt);

    //c
    //calculate y with forward Euler
    y_forwardE = y_old_f + (v_old*dt);

    //e
    //calculate leap frog method (averaging every step)
    v_avg = (v + v_old)/2;
    y_leapfrog = y_old_l + (v_avg * dt);

    //f
    //calculate leapfrog with backward Euler with rewinding velocity
    v_lf = v_old_lf + (g * (dt));
    y_lf = y_old_lf + (v_lf * dt);

    //calculate analytical solution
    y_analytical = 0.5 * (g * t * t) + y0;
```


Part 4 b and c (10 iterations)
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
Forward
Backward
Analytical


50 iterations
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
0.1
0.2
0.3
0.4
0.5
0.6
forward
backward
analytical



I modified the code to set the initial velocity back a half step and then continue with the iteration using the Backward Eulerian method and confirmed it matched the leapfrog data where the velocity is averaged every iteration.