Homework 2

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
plt.style.use('seaborn-whitegrid')
```

In [ ]:
```
```

Problem 1

##
##

The velocity (v) of an electric bicycle with the power output (P) of the motor is described by the following differential equation:

$$\frac{dv}{dt} = \frac{P}{mv}$$

where, m is total mass (bicycle + rider)

Given:
v(0) = 4.0 m/s
P = 400 Watt
m = 70 kg
Δt (or h) = 0.1 sec
Solve the v(t) by Euler method for time interval [0, 200].
Compare the result with the exact solution,

$$v = \sqrt{v(0)^2 + 2Pt/m}$$

(plot v against t).
(you may vary time step to see the accuracy variation)

##
##

Solution Problem 1

First order approximation by Euler's method,

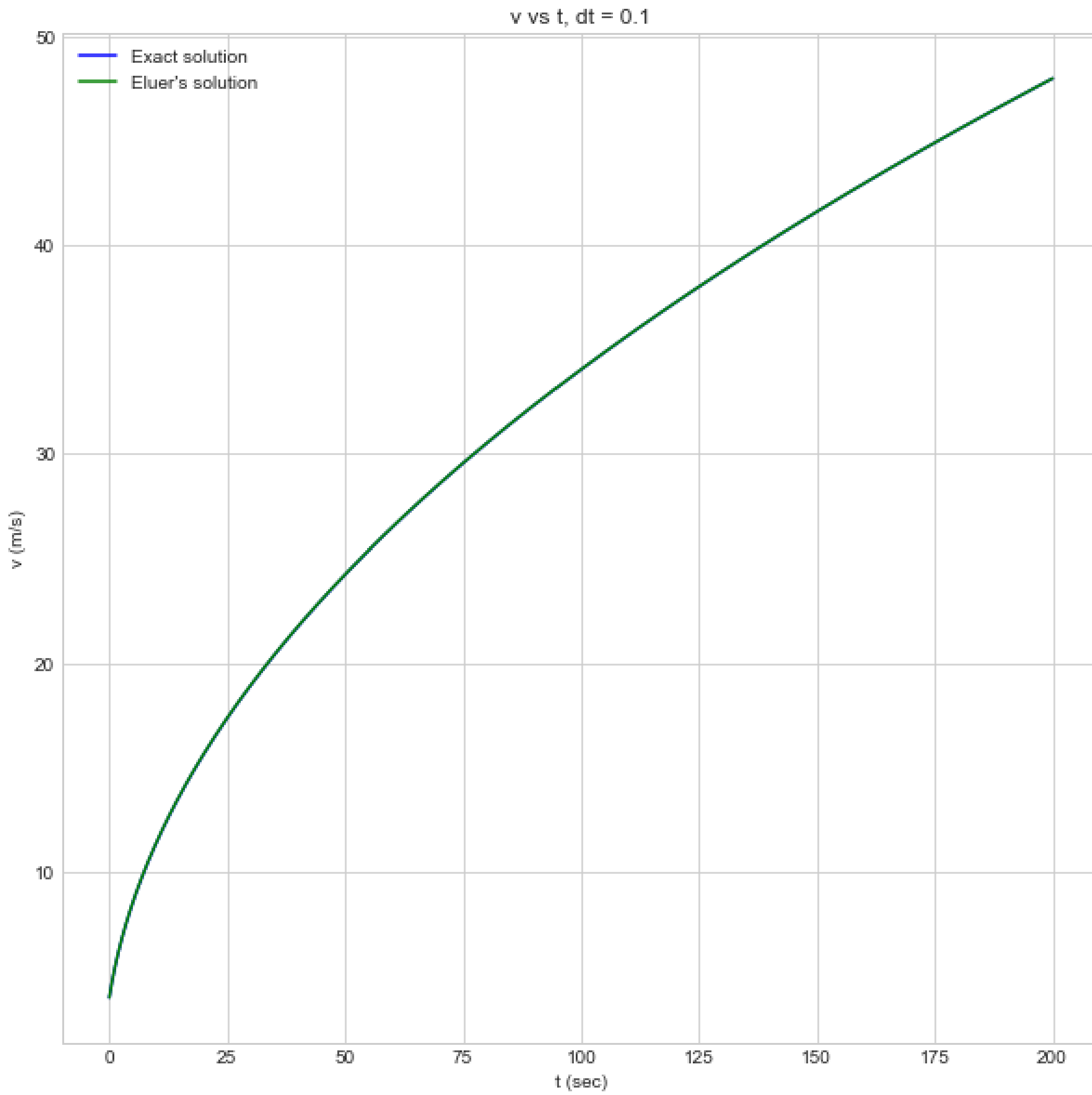
$$x(t+\Delta t) \approx x(t) + \frac{dx}{dt}\Delta t = x(t) + f(x(t),t)\Delta t$$

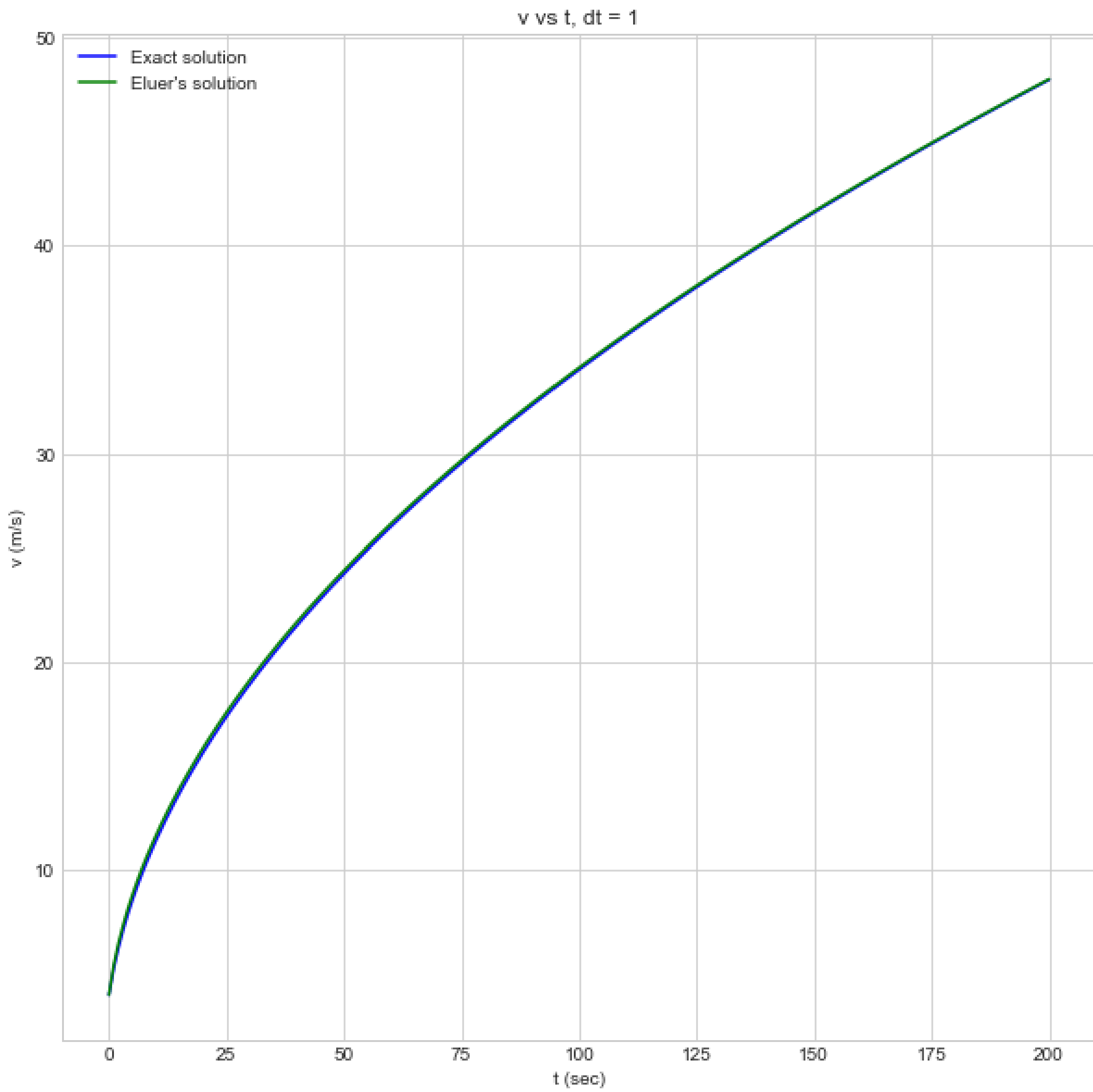
In [2]:
```python
################################################################################
######################## euler #################################################
def euler(f, a, b, dt, x_0, x_exact = None):

    """
    General info:
        This function returns solutions to the dependent variable (x's) of a
        differential equation for the given range of independent variable (t).
    Arguments:
        f       : 1st derivative of function
        a       : Lower limit
        b       : Higher limit
        dt      : Time step
        x_0     : x at time 0 in m/s
        x_exact : Exact solution (for plot comparision)
    """

    t    = np.linspace(a, b, int((b-a)/dt))    # >> Independent variable t
    x    = [0] * int((b-a)/dt)                 # >> Dependent variable x
    x[0] = x_0

    # Loop for getting x's >>
    for i in range(len(t) - 1):
        x[i+1] = x[i] + f(x[i], t[i]) * dt

    # Plotting results >>
    fig  = plt.figure(figsize = (10, 10))
    axes = plt.gca()
    if x_exact: axes.plot( t, x_exact(t),        # >> Plotting exact solution
                           color     = "blue",
                           label     = "Exact solution" )
    axes.plot(             t, x,                 # >> Plotting euler's solution
                           color     = "green",
                           label     = "Eluer's solution" )

    # Setting plot elements >>
    label_xaxis = "t"
    label_yaxis = "v"
    title       = label_yaxis + " vs " + label_xaxis + ", dt = " + str(dt)
    axes.set_title(title)
    axes.set_xlabel(label_xaxis + " (sec)")
    axes.set_ylabel(label_yaxis + " (m/s)")
    axes.legend()

    return None
######################## euler #################################################
################################################################################
```

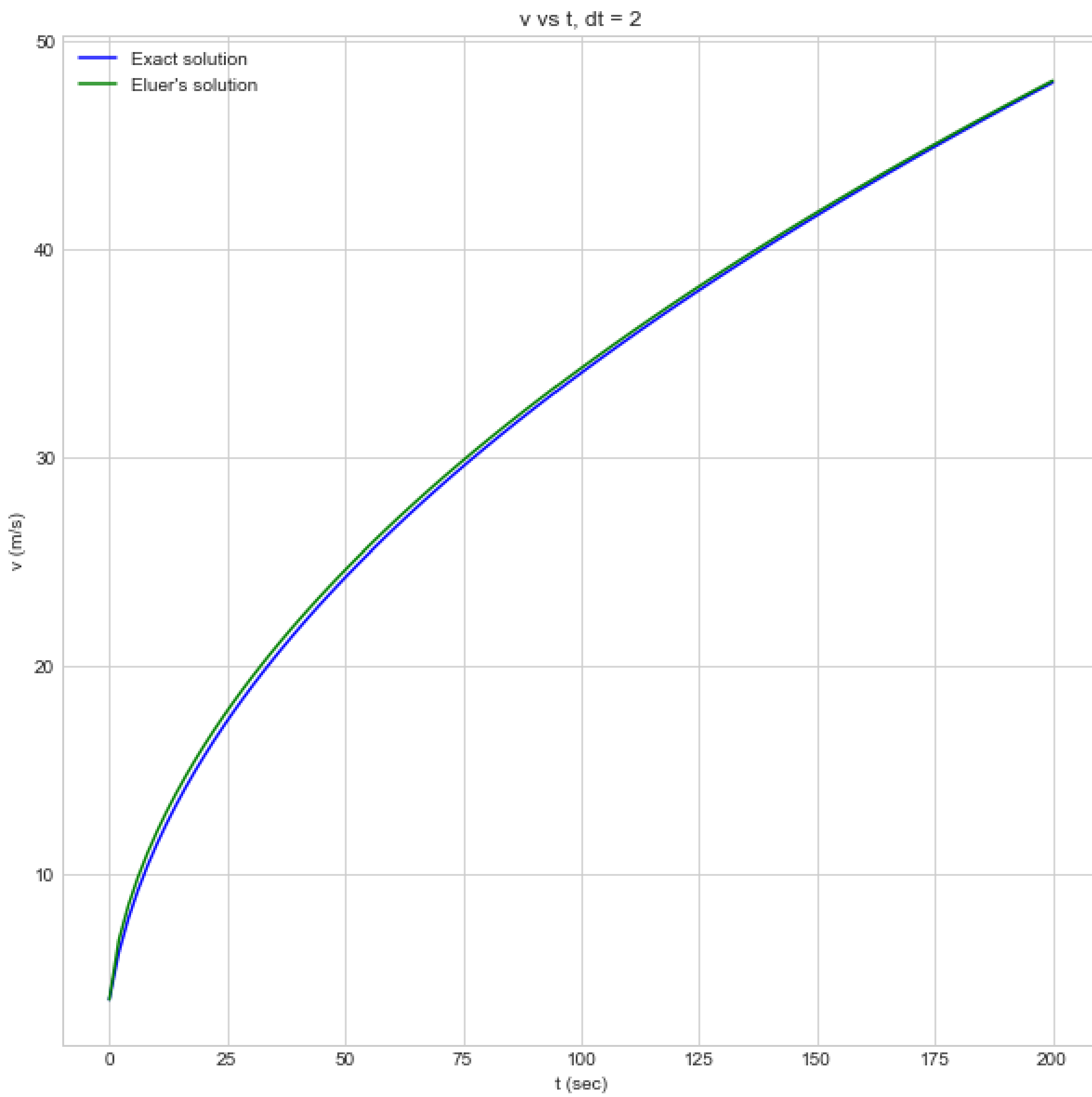
```
################################################################################
######################## Given ################################################
P   = 400 # >> Power output of motor in Watts
m   = 70  # >> Total mass (bicycle + rider) in kg
######################## Given ################################################
################################################################################


################################################################################
######################## der_1st ##############################################
def der_1st(v, t):
    return( P / (m*v) )
######################## der_1st ##############################################
################################################################################


################################################################################
######################## v_exact ##############################################
def v_exact(t):
    v_0 = 4                  # >> Velocity at time 0 in m/s
    return( np.sqrt( v_0**2 + 2*P*t/m ) )
######################## v_exact ##############################################
################################################################################
```

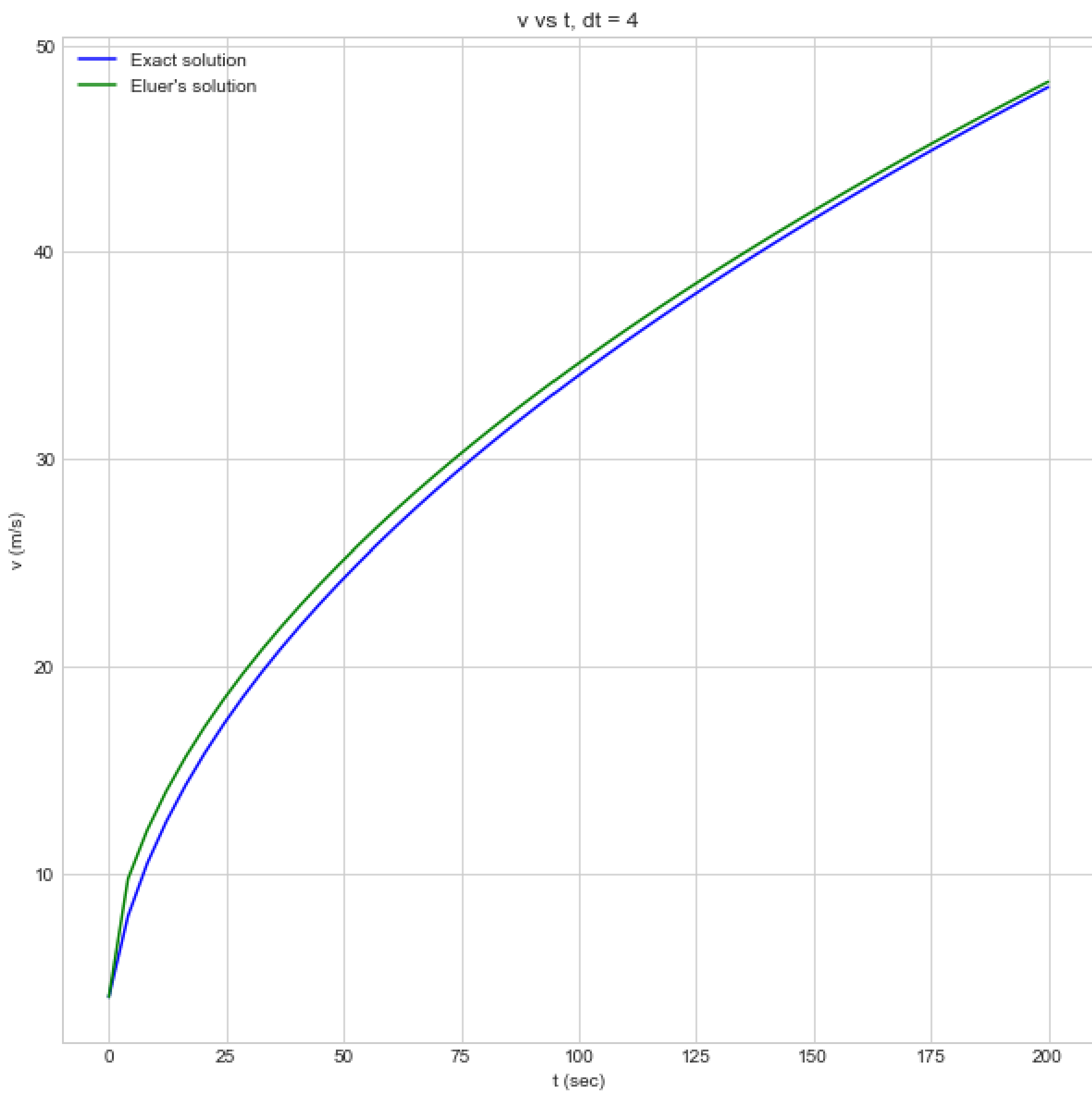
Final Solution Problem 1

In [3]:
```
for dt in [0.1, 1, 2, 4, 6, 8, 10]:
    euler( f       = der_1st,
           a       = 0,
           b       = 200,
           dt      = dt,
           x_0     = 4,
           x_exact = v_exact )
```

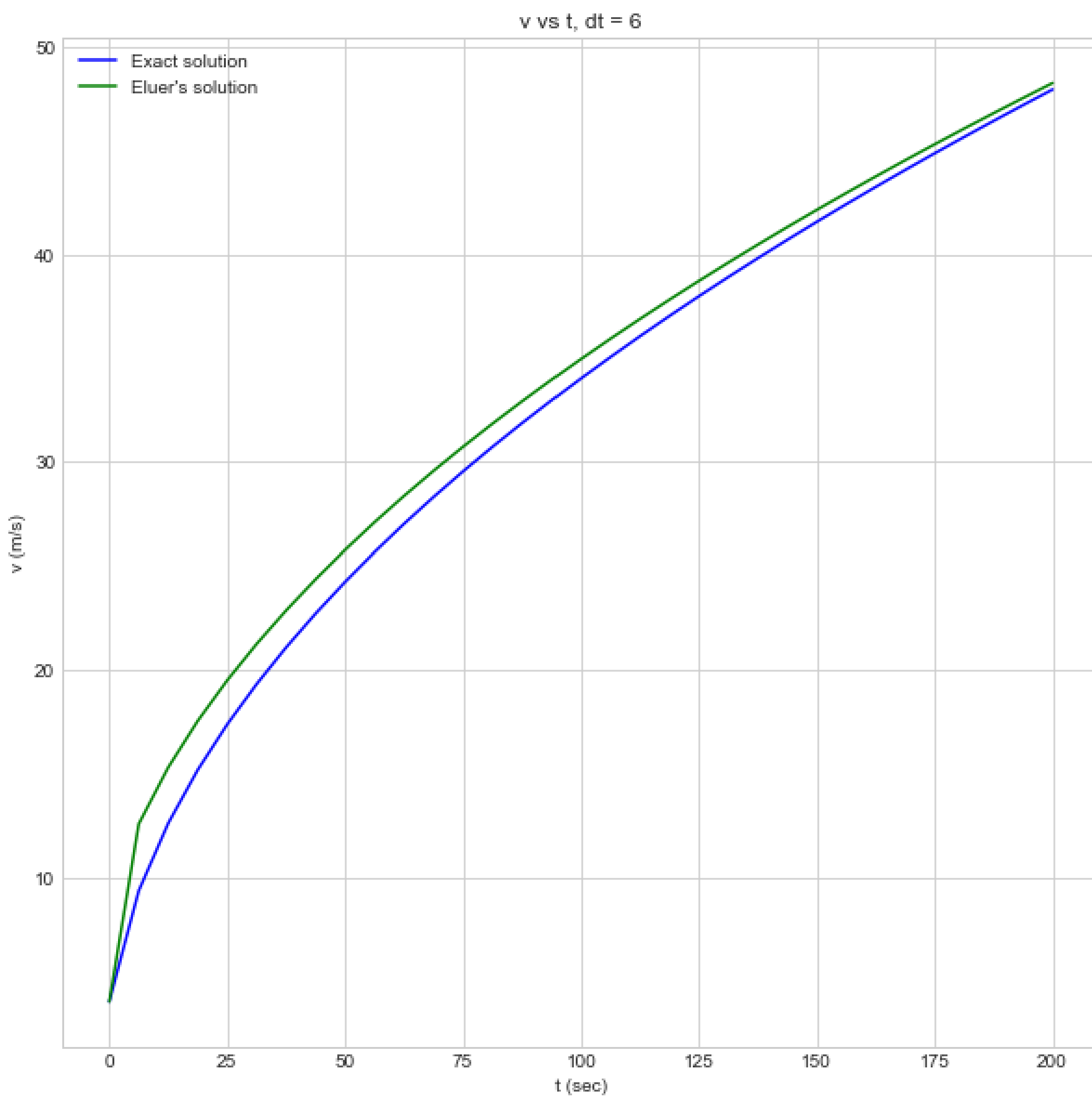


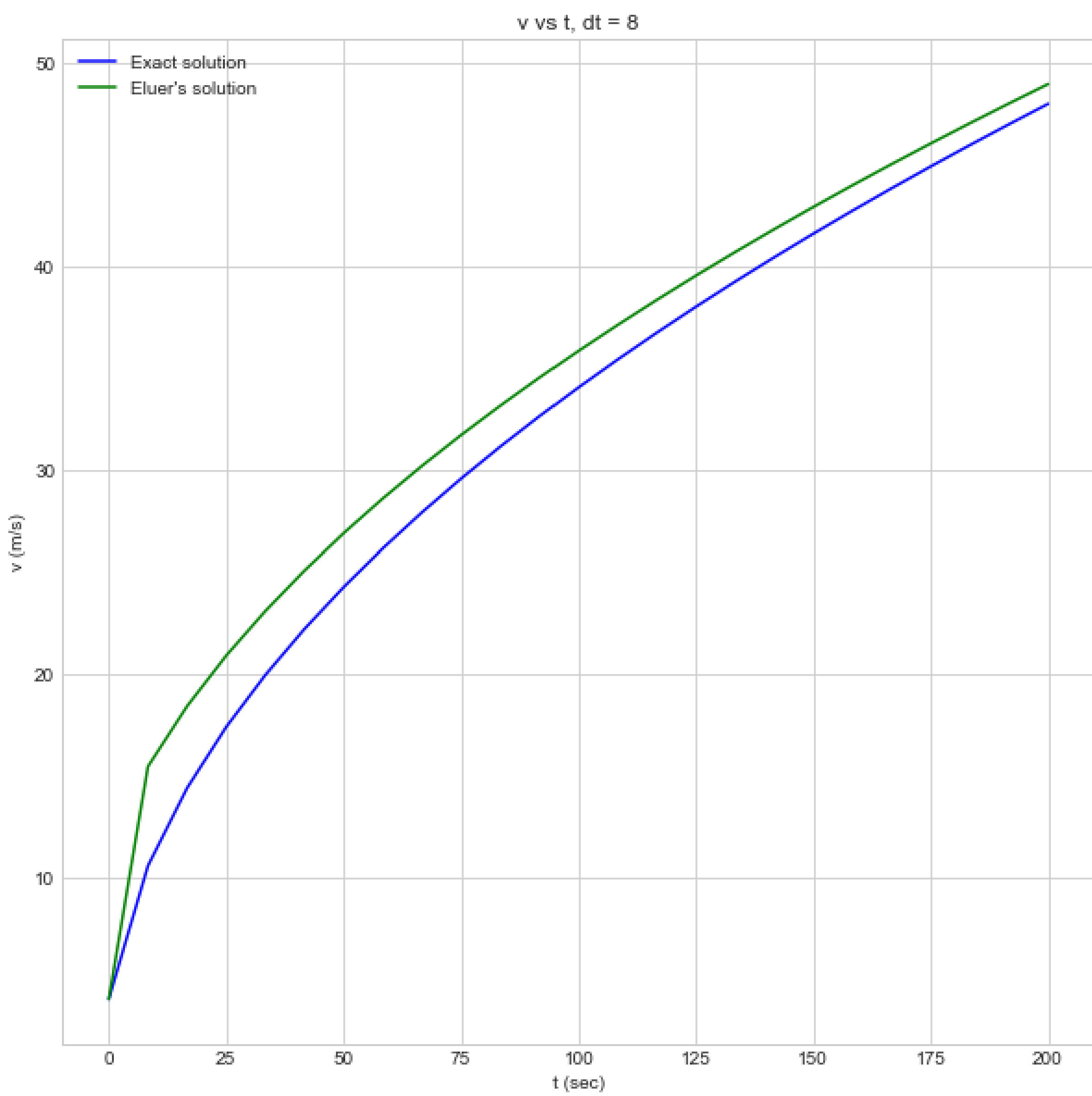

v vs t, dt = 1
Exact solution
Eluer's solution
v (m/s)
t (sec)
10
20
30
40
50
0
25
50
75
100
125
150
175
200


v vs t, dt = 2
Exact solution
Eluer's solution
v (m/s)
t (sec)
0
25
50
75
100
125
150
175
200
10
20
30
40
50

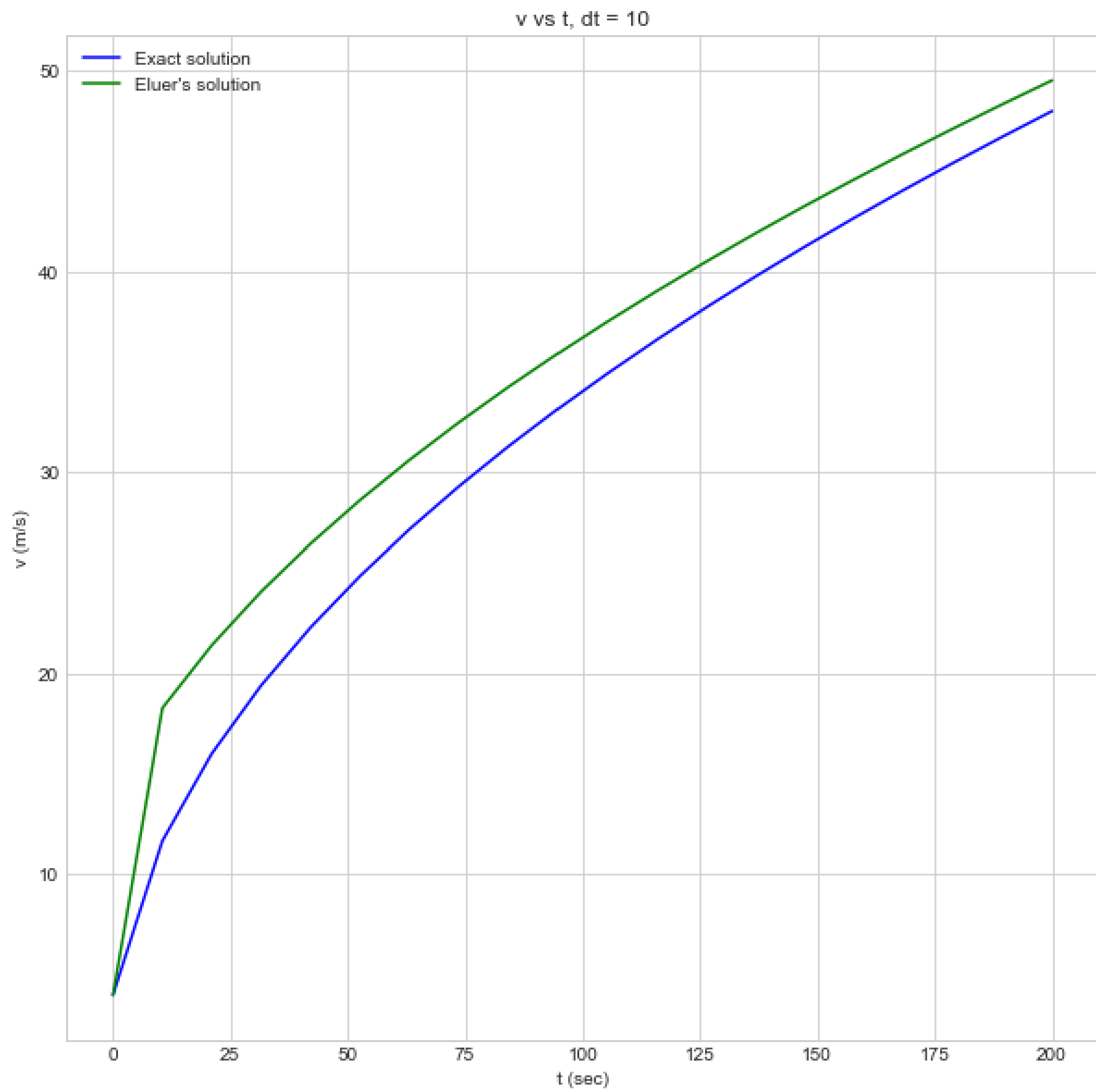


v vs t, dt = 4
Exact solution
Eluer's solution
v (m/s)
t (sec)
10
20
30
40
50
0
25
50
75
100
125
150
175
200


v vs t, dt = 6
Exact solution
Eluer's solution
v (m/s)
t (sec)
0
25
50
75
100
125
150
175
200
10
20
30
40
50



v vs t, dt = 8
Exact solution
Eluer's solution
v (m/s)
t (sec)
0
25
50
75
100
125
150
175
200
10
20
30
40
50



In [ ]:

In [ ]:

Problem 2

##
##

Add in an air-drag term, $-Av^2$, on the right-hand of the equation in Problem 1,
where A is proportional to the frontal area and the air density.
Then solve for v(t) using A = 0.15 and compare the result to the solution with A = 0.

##
##

Solution Problem 2

$$\frac{dv}{dt} = \frac{P}{mv} - Av^2$$

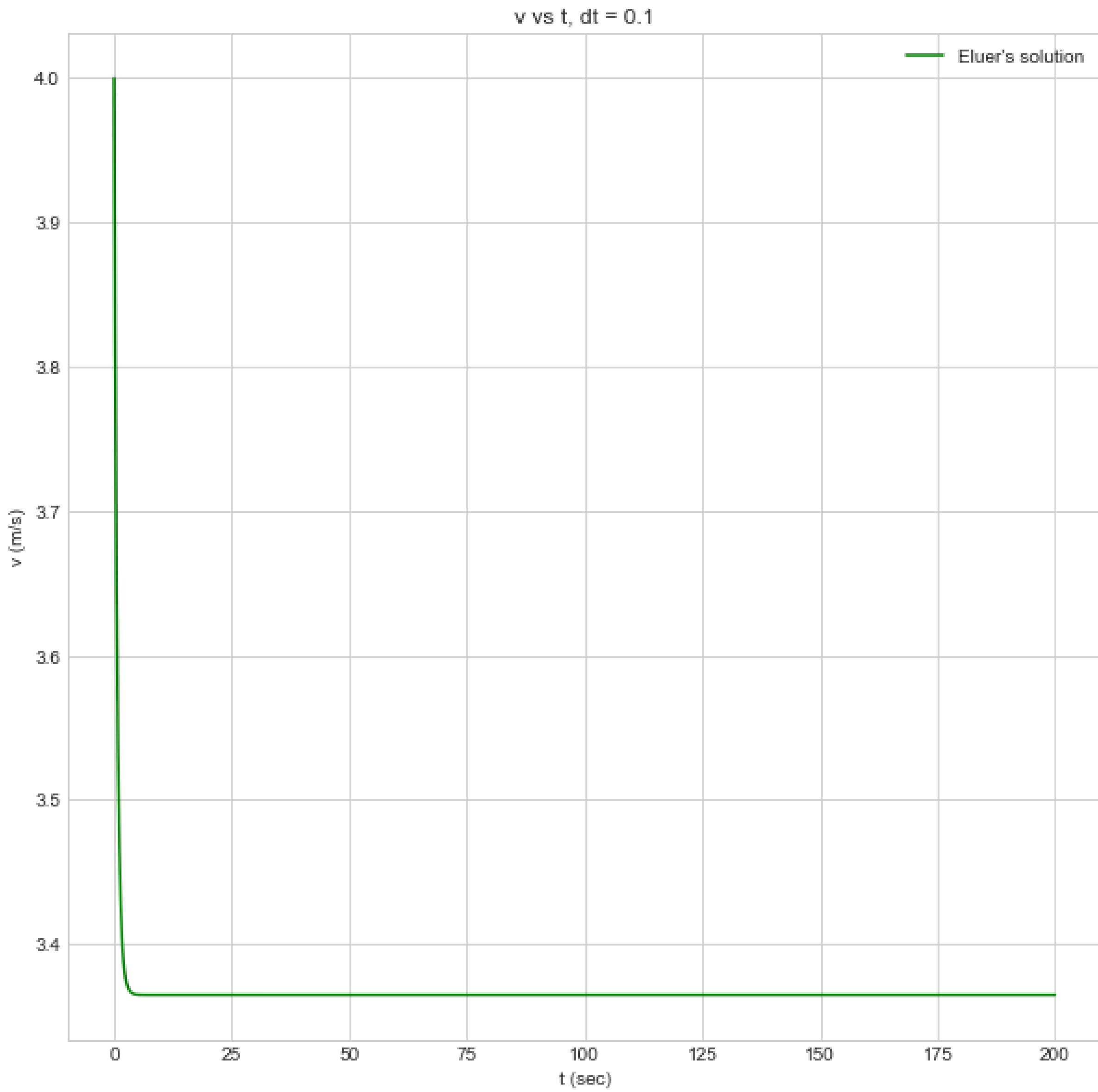
In [4]:

```
################################################################################
######################## der_1st_air_drag ######################################
A = 0.15
def der_1st_air_drag(v, t):
    return( P / (m*v) - A*v**2 )
######################## der_1st_air_drag ######################################
################################################################################
```

Final Solution Problem 2

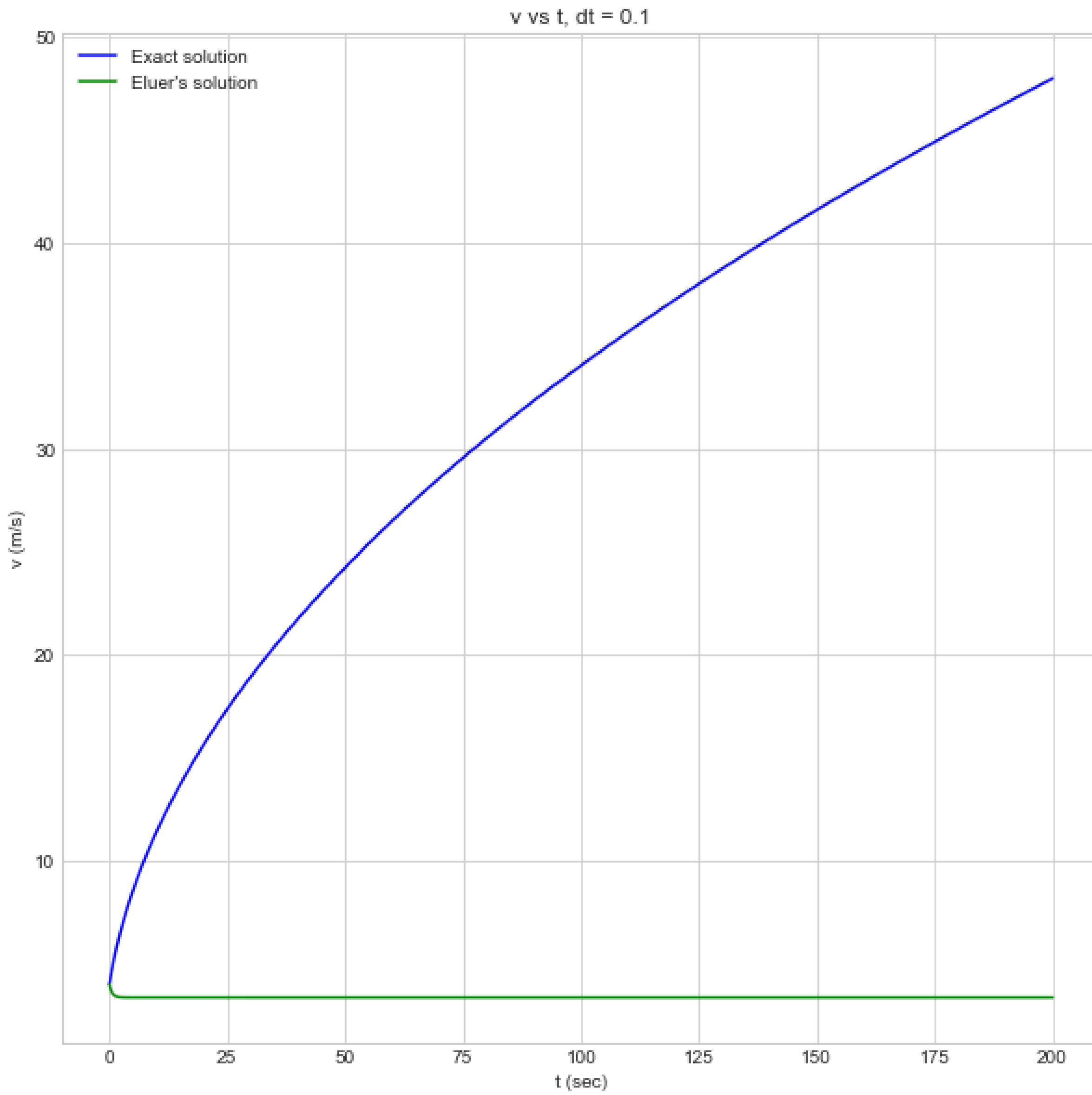
A = 0.15

In [5]:
```
euler( f        = der_1st_air_drag,
       a        = 0,
       b        = 200,
       dt       = 0.1,
       x_0      = 4
     )
```



A = 0.15 vs A = 0

In [6]:
```
euler( f       = der_1st_air_drag,
       a       = 0,
       b       = 200,
       dt      = 0.1,
       x_0     = 4,
       x_exact = v_exact
     )
```



In [ ]:

```

```

In [ ]:

```

```

Problem 3

##
##

Show that the Verlet algorithm to solve a 2nd-order ODE,

$$\frac{d^2x}{dt^2} = F(x,t)/m$$

has an error in the order of magnitude of $O(\Delta t^4)$:

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + [\frac{F(x(t))}{m}]\Delta t^2$$

Hint: using Taylor expansion for $x(t+\Delta t)$ and $x(t-\Delta t)$.

##
##

Final Solution Problem 3

Solution Problem 3

Taylor expansion for $x(t+\Delta t)$,

$$x(t+\Delta t) \approx x(t) + \frac{dx}{dt}\Delta t + \frac{1}{2}\frac{d^2x}{dt^2}\Delta t^2 \quad \text{——— (i)}$$

& Taylor expansion for $x(t-\Delta t)$,

$$x(t-\Delta t) \approx x(t) - \frac{dx}{dt}\Delta t + \frac{1}{2}\frac{d^2x}{dt^2}\Delta t^2 \quad \text{——— (ii)}$$

Now, (i) + (ii)

$$x(t+\Delta t) + x(t-\Delta t) = 2x(t) + \frac{d^2x}{dt^2}\Delta t^2$$

$$\therefore \boxed{x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \left[\frac{F(x(t))}{m}\right]\Delta t^2}$$

Error $O(\Delta t^4)$

In [ ]:

In [ ]:

In [ ]: