Question 2: Enzyme Kinetics

8.1 Let the concentrations of E = e, S = s, ES = c and P = p respectively.

$$\frac{de}{dt} = -k1(e)(s) + k2(c) + k3(c)$$

$$\frac{ds}{dt} = -k1(e)(s) + k2(c)$$

$$\frac{dc}{dt} = k1(e)(s) - k2(c) - k3(c)$$

$$\frac{dp}{dt} = k3(c)$$

8.2

$e_0 = 1uM$

$s_0 = 10uM$

$c_0 = 0uM$

$p_0 = 0uM$

$k1 = 100/uM/min$

$k2 = 600/min$

$k3 = 150/min$

The free and bound enzyme concentrations should add up to the total initial enzyme concentration.

$c + e = e_0$

$e = e_0 - c$

Code to solve the rate equations:

```
# Python programme to solve enzyme kinetics rate equations using Runge-Kutta 4th
order Method
# define differential equations
def dsdt(s, c):
    return ((-k1*(e0-c)*s) + k2*c)
def dcdt(s, c):
    return ((k1*(e0-c)*s)-(k2*c)-(k3*c))

# define Runge-Kutta 4th order method function with inital values of s0, c0 at
t0, step size = h, time of interest = t
def rungeKutta(s0, c0, t0, t, h):
    # count number of iterations needed
    n = round((t-t0)/h)
```

```
    # define initial s and c
    s = s0
    c = c0

    # set initial area under curve c vs t = 0 (for calculation of p)
    auc = 0

    # use loop to apply Runge-Kutta method to solve for s and c
    for i in range(n):
        ks1 = dsdt(s,c)
        kc1 = dcdt(s,c)

        sHalf = s + ks1*h/2
        cHalf = c + kc1*h/2

        if sHalf < 0:
            raise Exception("concentration of s went below 0, reduce step size or
time range")

        ks2 = dsdt(sHalf, cHalf)
        kc2 = dcdt(sHalf, cHalf)

        sHalf = s + ks2*h/2
        cHalf = c + kc2*h/2

        if sHalf < 0:
            raise Exception("concentration of s went below 0, reduce step size or
time range")

        ks3 = dsdt(sHalf, cHalf)
        kc3 = dcdt(sHalf, cHalf)

        sFull = s + ks3*h
        cFull = c + kc3*h

        if sFull < 0:
            raise Exception("concentration of s went below 0, reduce step size or
time range")

        ks4 = dsdt(sFull, cFull)
        kc4 = dcdt(sFull, cFull)

        cinitial = c
```

```
        # update values of s and c
        s = s + (1.0/6.0)*(ks1+2*ks2+2*ks3+ks4)*h
        c = c + (1.0/6.0)*(kc1+2*kc2+2*kc3+kc4)*h

        if s < 0:
            raise Exception("concentration of s went below 0, reduce step size or 
time range")

        # calculate approximate area under the curve c vs t using trapezoidal 
rule
        auc = auc + h/2*(c+cinitial)
    return s, c, auc


# to get solution
# define initial values and constants
t0 = 0
e0 = 1
s0 = 10
c0 = 0
p0 = 0
k1 = 100
k2 = 600
k3 = 150

# input time of interest t, and step size h
t = 0.001
h = 0.001

# execute rungeKutta function
(s,c,auc) = rungeKutta(s0, c0, t0, t, h)

# calculate e
e = e0 - c

# calculate p, where p = k3 * integral of c with respect to t (i.e area under the 
curve of c vs t)
p = k3 * auc

print(f"At time t= {t}, s= {s}, c= {c}, e= {e}, p= {p}")
```

8.3

$$V = \frac{dp}{dt} = k3(c)$$

Assuming equilibrium condition, c reaches a steady state, so that $\frac{dc}{dt} = 0$.

$$k1(e)(s) - k2(c) - k3(c) = 0$$

$$k1(e)(s) = (k2 + k3)(c)$$

$$k1(e_0 - c)(s) = (k2 + k3)(c)$$

$$\frac{(e_0 - c)(s)}{(c)} = \frac{(k2 + k3)}{k1}$$

$$\frac{(e_0 s - cs)}{(c)} = \frac{(k2 + k3)}{k1}$$

$$\frac{(e_0 s)}{(c)} - s = \frac{(k2 + k3)}{k1}$$

$$\frac{(e_0 s)}{(c)} = \frac{(k2 + k3)}{k1} + s$$

$$c = \frac{(e_0 s)}{\frac{(k2 + k3)}{k1} + s}$$

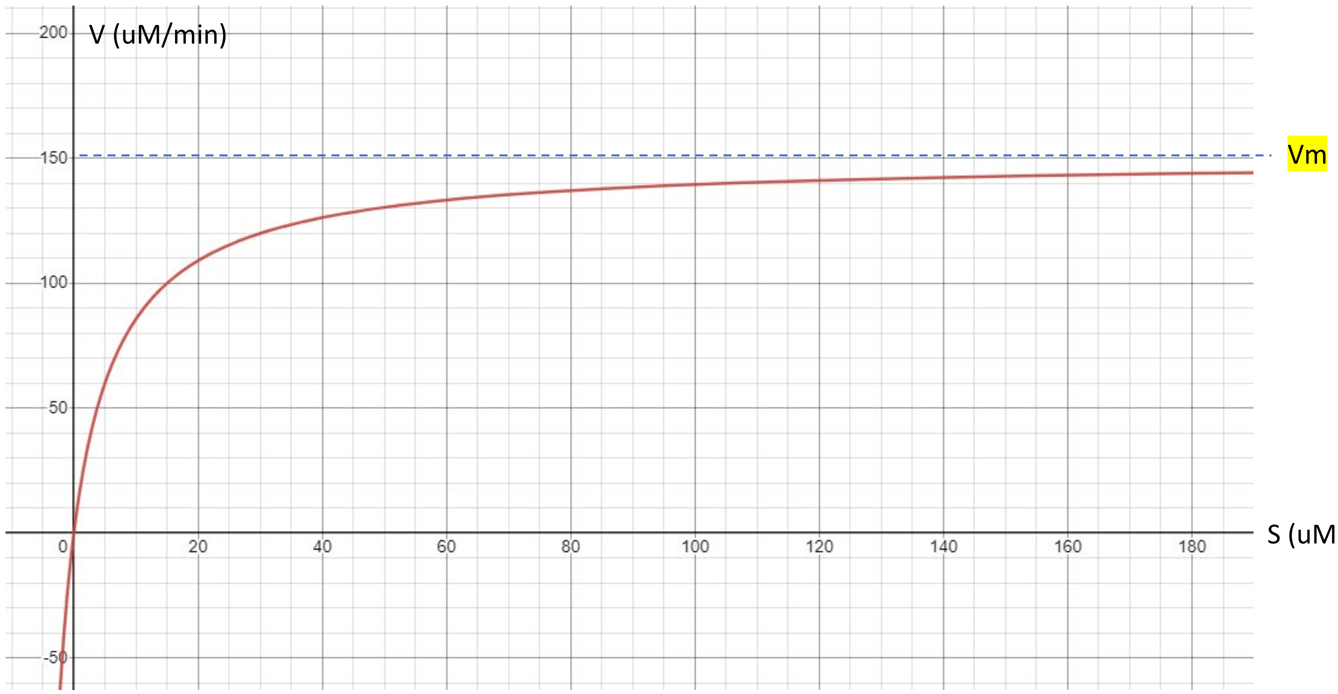
$$V = k3(c)$$

$$V = k3\left(\frac{(e_0 s)}{\frac{(k2 + k3)}{k1} + s}\right)$$

$V = \frac{Vm(s)}{Km+s}$, where $Vm = k3(e_0)$, $Km = \frac{(k2+k3)}{k1}$

$$Vm = (150)(1) = 150uM/min$$

$$Km = \frac{600 + 150}{100} = 7.5uM$$


V (uM/min)
200
150
100
50
0
-50
20
40
60
80
100
120
140
160
180
S (uM)
Vm