Assignment 2

1. Script for (a) + (b):

```
Editor - C:\Users\Titus\Documents\MATLAB\New Folder\Assignment 2\assignment2qn1.m
assignment2qn1.m
%part a
syms y(t) k
%define ode
ode = diff(y,t) == k*y;
%solve for general equation of y in t
general = dsolve(ode)
%set up equation with the statement: bacteria doubles every 5 hours
y1 = general;
y2 = subs(general,t+5);
eqn = y2/y1 == 2;
%solve for k
value = double(solve(eqn,k))

%part b
%obtain equation in y with values
cond = y(0) == 1;
y = dsolve(ode,cond)
y = subs(y,k=value)

%calculate number of bacteria from t = 0 to 12
A = [];
for t = [0:12]
    number = double(subs(y,t));
    A = [A; number];
end
bacteriaNumber = A

%graph plot
fplot(y,[0 12],'-ob')
grid
title('Number of bacteria vs time (hours)')
xlabel('time t (hours)')
ylabel('Number of bacteria present in culture')
```

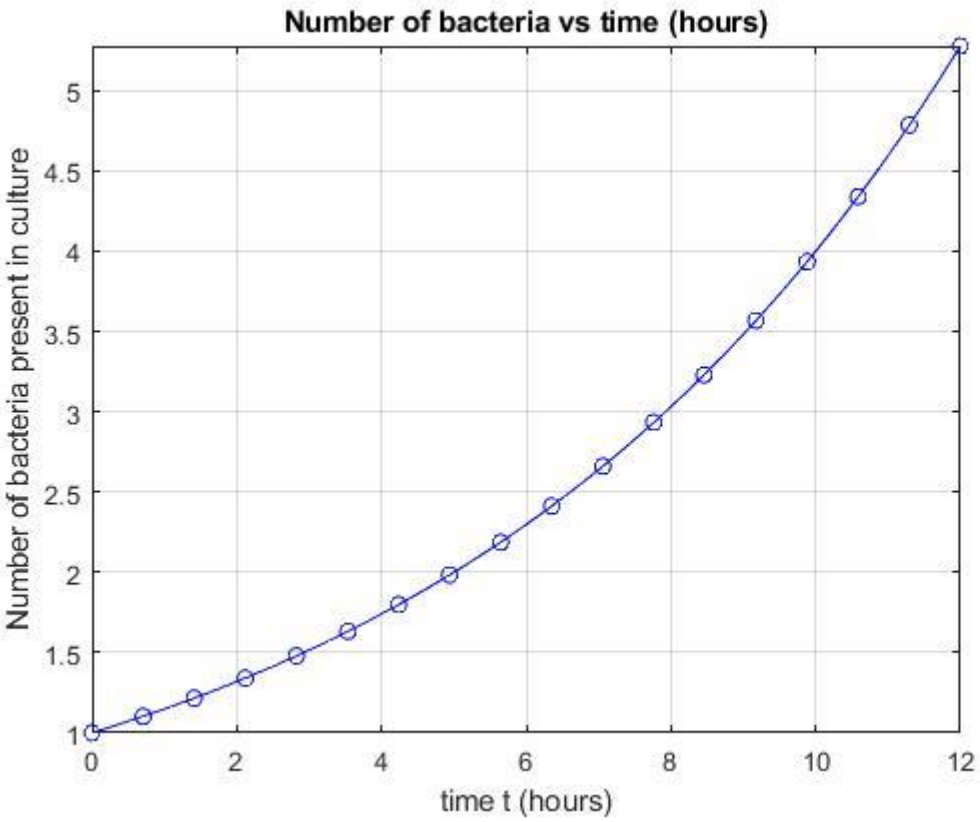
a)

```
Command Window
>> run assignment2qn1.m

general =

C1*exp(k*t)


value =

    0.1386


y =

exp(k*t)


y =

exp((4994651814532287*t)/36028797018963968)

>> 
```

From above results, value = k = 0.1386.

b) Bacteria Number for time $t \in [0,12]$

```
bacteriaNumber =

    1.0000
    1.1487
    1.3195
    1.5157
    1.7411
    2.0000
    2.2974
    2.6390
    3.0314
    3.4822
    4.0000
    4.5948
    5.2780
```



2. Scripts:

Ladder.m (function representing length of the ladder in terms of x):

Editor - C:\Users\Titus\Documents\MATLAB\New Folder\Assignment 2\ladder.m

tut2qe.m | tutorial2.m | assignment2q2.m | ladder.m

```
function f = ladder(x)
f = sqrt(x.^2+16)+sqrt(196+16.*x.^2)./x;
end
```

Editor - C:\Users\Titus\Documents\MATLAB\New Folder\Assignment 2\assignment2q2.m

tut2qe.m | tutorial2.m | assignment2q2.m | ladder.m

```
%check unimodal function for sufficiently large interval of x
fplot(@ladder,[0,100])
%minimum found between 0 to 10
%perform minimisation at a selected interval
x = fminbnd(@ladder,0,10,optimset('Display','iter'))
%obtain minimum ladder length
L = ladder(x)
```

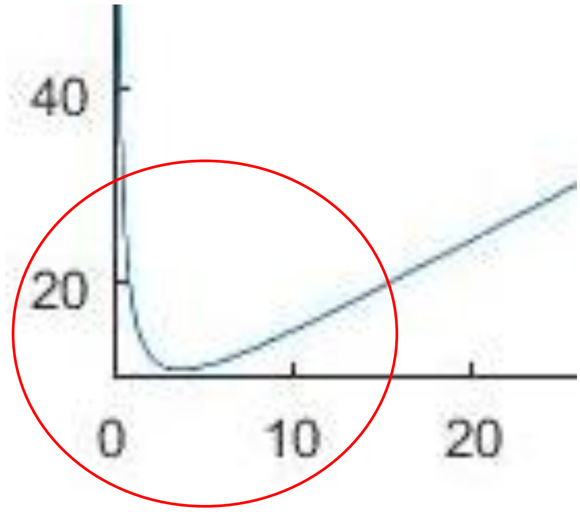
Results:

Command Window

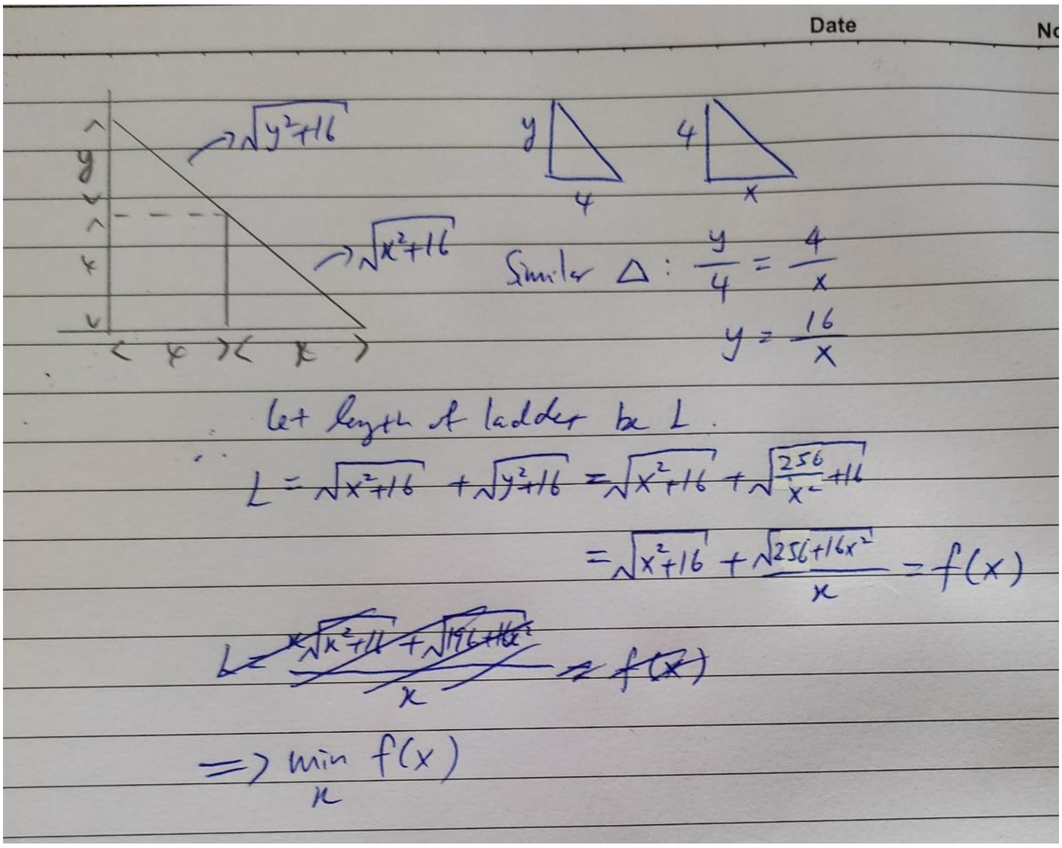
```
>> run assignment2q2.m

 Func-count     x          f(x)         Procedure
    1        3.81966      10.9561        initial
    2        6.18034      11.9587        golden
    3        2.36068       11.798        golden
    4        4.19031      11.0048        parabolic
    5        3.83549      10.9569        parabolic
    6        3.73023      10.9545        parabolic
    7        3.20711      11.0477        golden
    8        3.74094      10.9545        parabolic
    9        3.74178      10.9545        parabolic
   10        3.74166      10.9545        parabolic
   11        3.74169      10.9545        parabolic
   12        3.74162      10.9545        parabolic

Optimization terminated:
 the current x satisfies the termination criteria using OPTIONS.TolX of 1.000000e-04


x =

    3.7417


L =

   10.9545
```

Thus, the minimum ladder length L = 10.95m.

Cropped graph checking unimodal Function for x > 0:



Finding the ladder function to be minimized:



Similar Δ: $\frac{y}{4} = \frac{4}{x}$

$y = \frac{16}{x}$

Let length of ladder be L.

$\therefore L = \sqrt{x^2+16} + \sqrt{y^2+16} = \sqrt{x^2+16} + \sqrt{\frac{256}{x^2}+16}$

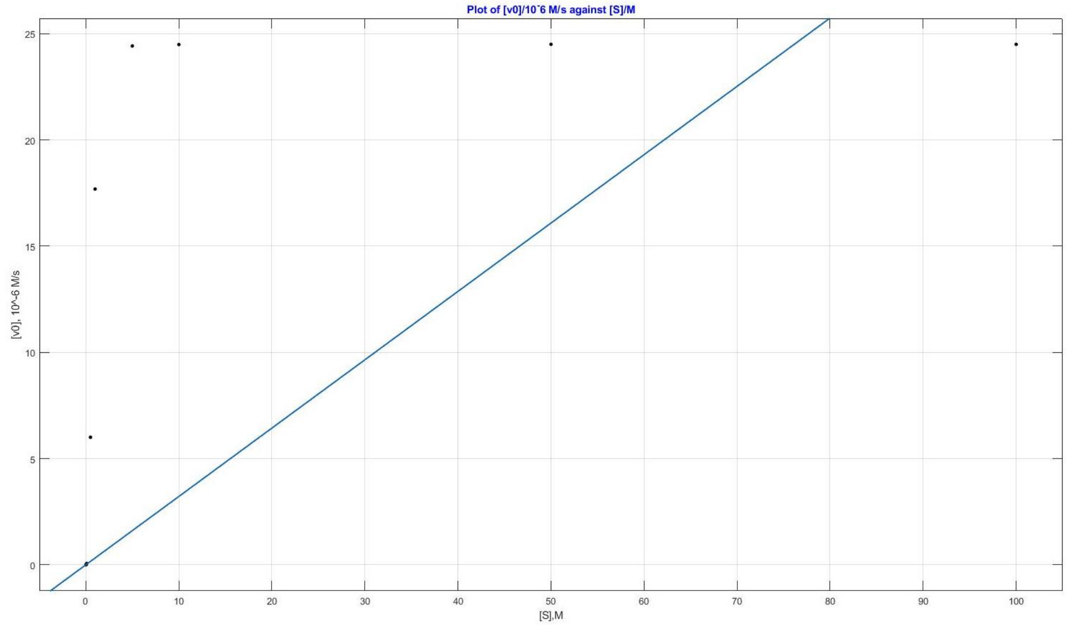
$= \sqrt{x^2+16} + \frac{\sqrt{256+16x^2}}{x} = f(x)$

$\cancel{L = \frac{x\sqrt{x^2+16} + \sqrt{256+16x^2}}{x} = f(x)}$

$\Rightarrow \min_x f(x)$

3.

a) Plot of v_0 against $[S]$ (Force intercept term to be 0):



Results

Linear model Poly1:
f(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
p1 = 0.3218 (0.04657, 0.597)
p2 = 0 (fixed at bound)

Goodness of fit:
SSE: 1439
R-square: -0.307
Adjusted R-square: -0.307
RMSE: 13.41

Warning: A negative R-square is possible if the model does not contain a constant term and the fit is poor (worse than just fitting the

Equation:

$$v_0 = 0.3218[S]$$

Values used:

Documents ▸ MATLAB ▸ New Folder ▸ Assignment 2

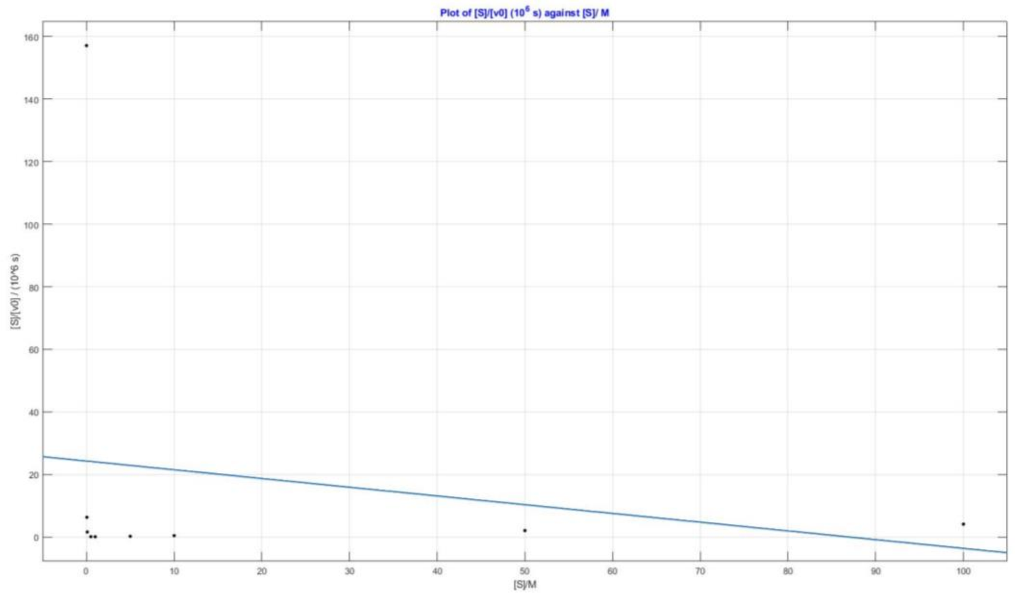
Command Window

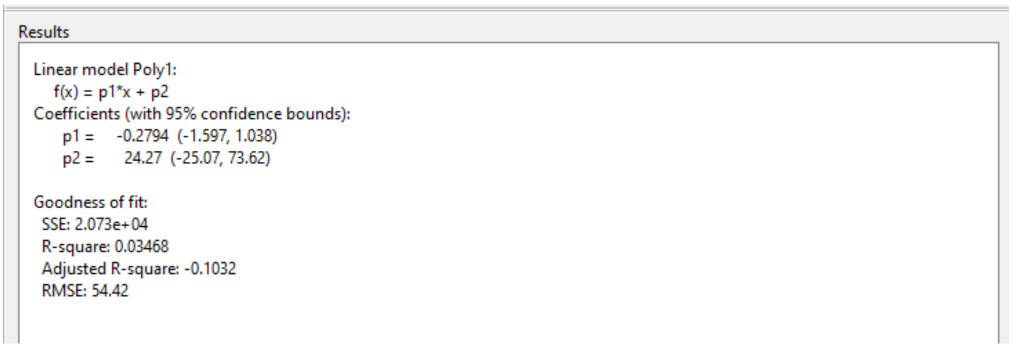
```
>> [S] = [0.01 0.05 0.1 0.5 1 5 10 50 100];
>> [v0] = [6.3636E-5 0.007952 0.063472 6.0049 17.69 24.425 24.491 24.5 24.5];
>> 
```

b) Rearranging $v_0 = \frac{k[S]}{K+[S]}$:

$$\frac{[S]}{v_0} = \frac{K}{k} + \frac{[S]}{k}$$

Plot of $\frac{[S]}{v_0}$ against $[S]$:




Results

Linear model Poly1:
f(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
p1 = -0.2794 (-1.597, 1.038)
p2 = 24.27 (-25.07, 73.62)

Goodness of fit:
SSE: 2.073e+04
R-square: 0.03468
Adjusted R-square: -0.1032
RMSE: 54.42

$$k = -3.5791, K = -86.865$$

Equation:

$$v_0 = -\frac{3.58[S]}{[S] - 86.9}$$

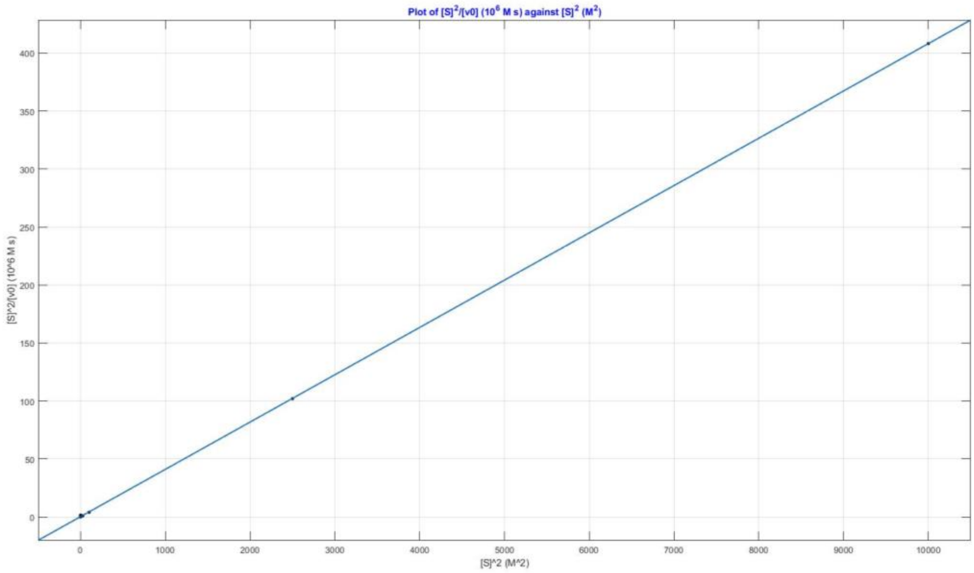
Values:

```
Command Window
>> [Y]

Y =

  Columns 1 through 8

  157.1438    6.2877    1.5755    0.0833    0.0565    0.2047    0.4083    2.0408

  Column 9

    4.0816

>> [S]

S =

  Columns 1 through 8

    0.0100    0.0500    0.1000    0.5000    1.0000    5.0000   10.0000   50.0000

  Column 9

  100.0000
```

c) Rearranging $v_0 = \frac{k[S]^2}{K+[S]^2}$:

$$\frac{[S]^2}{[v_0]} = \frac{K}{k} + \frac{[S]^2}{k}$$

Plot of $\frac{[S]^2}{[v_0]}$ against $[S]^2$:



Results

Linear model Poly1:
f(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
p1 = 0.04078 (0.04065, 0.04092)
p2 = 0.2793 (-0.1833, 0.7419)

Goodness of fit:
SSE: 2.009
R-square: 1
Adjusted R-square: 1
RMSE: 0.5357

$$k = 24.522, K = 6.8490$$

Equation:

$$v_0 = \frac{24.5[S]^2}{[S]^2 + 6.85}$$

Values:

Command Window

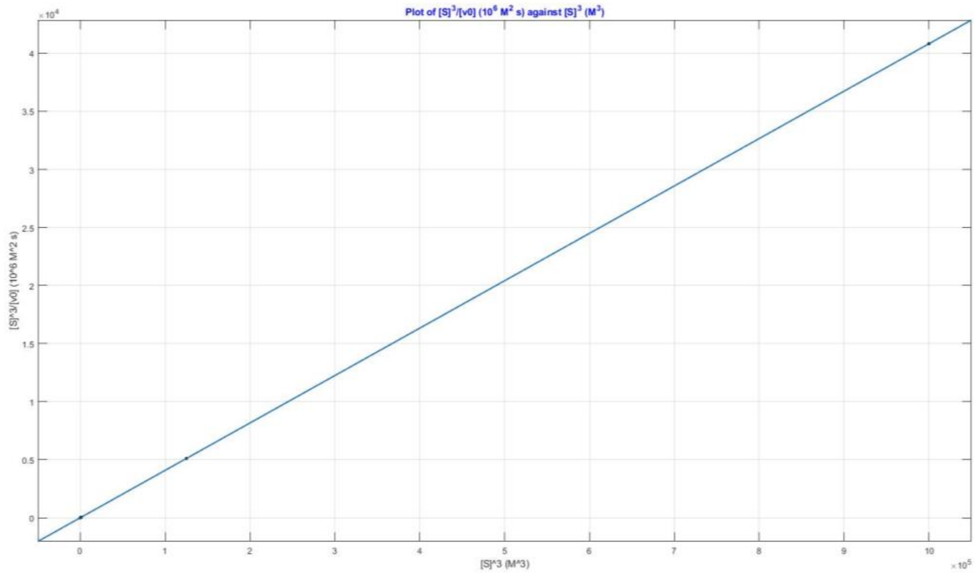
```
>> [Y1] = ([S].^2)./[v0]

Y1 =

  Columns 1 through 8

    1.5714    0.3144    0.1575    0.0416    0.0565    1.0235    4.0831  102.0408

  Column 9

  408.1633

>> [X1] = [S].^2

X1 =

   1.0e+04 *

  Columns 1 through 8

    0.0000    0.0000    0.0000    0.0000    0.0001    0.0025    0.0100    0.2500

  Column 9

    1.0000
```

First 4 x values: $1 \times 10^{-4}, 0.0025, 0.1, 0.25$.

d) Rearranging $v_0 = \frac{k[S]^3}{K+[S]^3}$:

$$\frac{[S]^3}{[v_0]} = \frac{K}{k} + \frac{[S]^3}{k}$$

Plot of $\frac{[S]^3}{[v_0]}$ against $[S]^3$:



Results

Linear model Poly1:
f(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
p1 = 0.04082 (0.04082, 0.04082)
p2 = 0.01409 (0.009952, 0.01823)

Goodness of fit:
SSE: 0.0001662
R-square: 1
Adjusted R-square: 1
RMSE: 0.004873

$$k = 24.498, K = 0.34518$$

Equation:

$$v_0 = \frac{24.5[S]^3}{[S]^3 + 0.345}$$

Values:

```
Command Window
>> [Y2] = ([S].^3)./[v0]

Y2 =

   1.0e+04 *

  Columns 1 through 8

    0.0000    0.0000    0.0000    0.0000    0.0000    0.0005    0.0041    0.5102

  Column 9

    4.0816

>> [X2] = [S].^3

X2 =

   1.0e+06 *

  Columns 1 through 8

    0.0000    0.0000    0.0000    0.0000    0.0000    0.0001    0.0010    0.1250

  Column 9

    1.0000
```

First 5 y values: $0.015714, 0.015719, 0.015755, 0.020816, 0.056529$

First 5 x values: $1 \times 10^{-6}, 1.25 \times 10^{-4}, 0.001, 0.125, 1$

Conclusion

All equations that are non-linear have been rearranged to perform linear fitting using the MATLAB curve fitting toolbox.

By visual inspection, the models yielding equations $v_0 = 0.3218[S]$ and $v_0 = -\frac{3.58[S]}{[S]-86.9}$ fits the data poorly as the points are far away from the fitted lines. The high SSE values of 1439 and 20,740 respectively also indicates the large distance between the data points and the fitted lines. In addition, both models yield r^2 values of -0.307 and 0.03468 respectively, which indicates that these models fit the experimental data poorly.

In contrast, models with equations $v_0 = \frac{24.5[S]^2}{[S]^2+6.85}$ and $v_0 = \frac{24.5[S]^3}{[S]^3+0.345}$ fits the data well according to the plots obtained. The data points in black are much closer to the fitted lines, indicated by the small SSE values of 2.009 and 0.0001662 respectively. From the software, both models have an r^2 value of 1, most likely rounded off since an r^2 = 1 is not possible in practice.

From the results, the model $v_0 = \frac{k[S]^3}{K+[S]^3}$ is the best for the current set of experimental data. Apart from having a very high r^2 value, the fitted equation has the smallest SSE value among the 4 models. The latter shows that the data points are very close to the fitted linear line, obtained from the rearranged model equation.

4. For a = 2, b = 1:

$$\left|\frac{x}{2}\right|^n + |y|^n \leq 1$$

Since the superellipse is symmetrical about either the x or y axes, area S = 4 times the area covered by the superellipse in 1 quadrant of the x-y plane about the origin.

For the quadrant where x, y > 0:

$$y = n\sqrt{1-\left(\frac{x}{2}\right)^n}$$

<u>Numerical Solution</u>

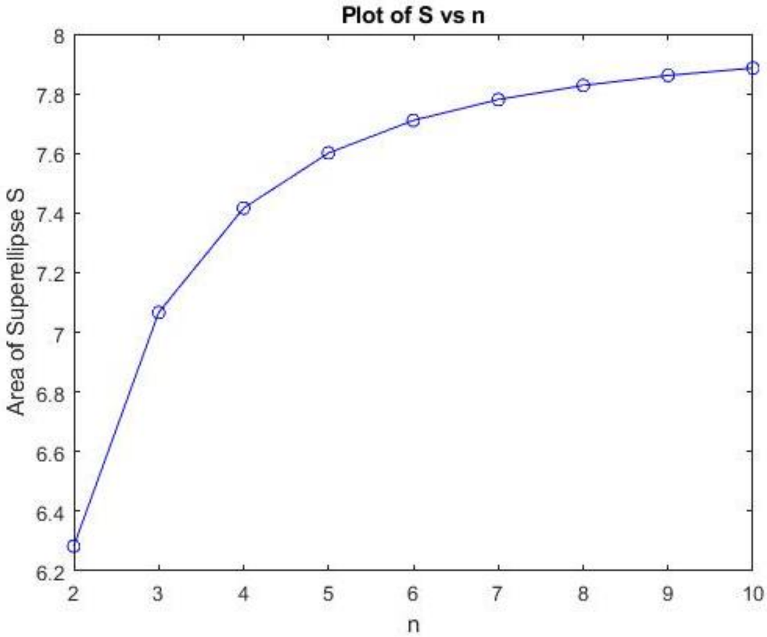
Script:

```
Editor - C:\Users\Titus\Documents\MATLAB\New Folder\Assignment 2\qn4numerical.m
qn4analytical.m   qn4numerical.m   +

areas = [];
f = @(x,n) nthroot((1-(x./2).^n),n);
for n = [2:10]
    area = integral(@(x) f(x,n),0,2).*4;
    areas = [areas; area];
end
S = areas
plot([2:10],S,'-ob')
title('Plot of S vs n')
xlabel('n')
ylabel('Area of Superellipse S')
```

The areas S for $n \in [2,10]$:

```
Command Window
>> run qn4numerical.m

S =

    6.2832
    7.0666
    7.4163
    7.6012
    7.7105
    7.7803
    7.8277
    7.8612
    7.8859

fx >>
```

Plot of S vs n, for $n \in [2,10]$:



<u>Analytical Solution</u>

Script:

Editor - C:\Users\Titus\Documents\MATLAB\New Folder\Assignment 2\qn4analytical.m

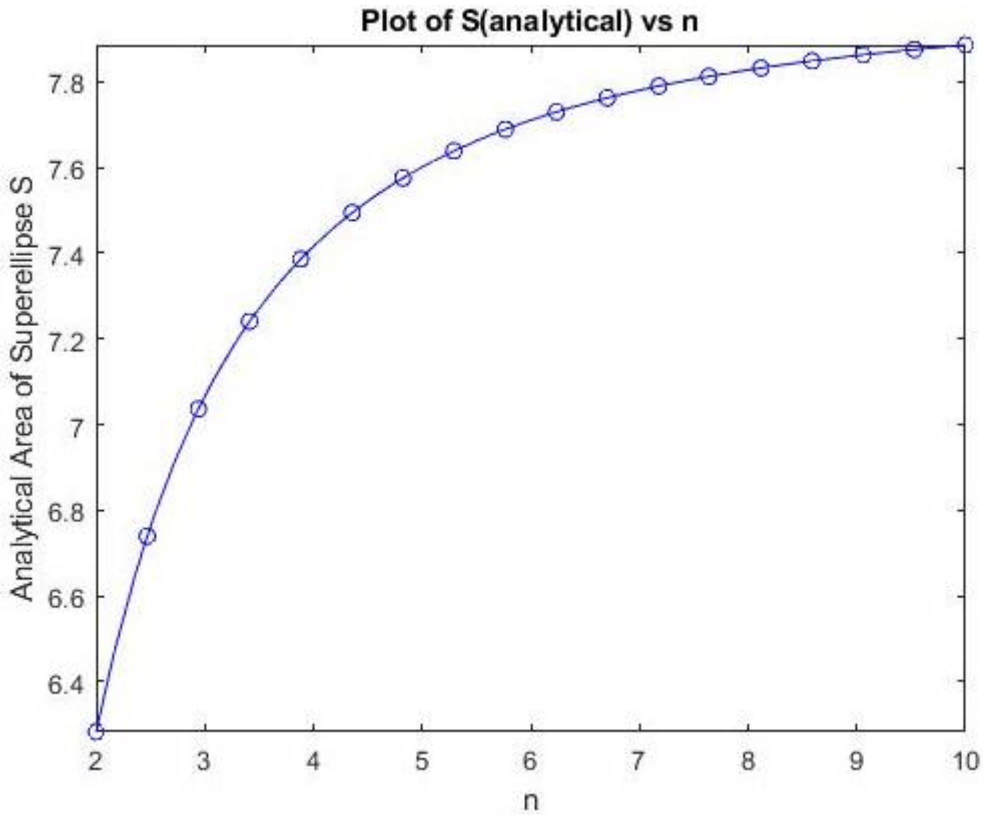
qn4analytical.m | qn4numerical.m | +

```
syms n
areas = [];
f = (4.^(1-1./n).*2.*sqrt(pi).*gamma(1+1./n))./gamma(0.5+1./n);
for n = [2:10]
    area = double(subs(f,n));
    areas = [areas; area];
end
Sa = areas
fplot(f,[2 10],'-ob')
title('Plot of S(analytical) vs n')
xlabel('n')
ylabel('Analytical Area of Superellipse S')
```

Analytical areas Sa for $n \in [2,10]$:

```
Command Window
>> run qn4analytical.m

Sa =

    6.2832
    7.0666
    7.4163
    7.6012
    7.7105
    7.7803
    7.8277
    7.8612
    7.8859
```

Plot of S vs n for $n \in [2,10]$:



<u>Comparison</u>

For both analytical and numerical solutions, the areas S are evaluated at integer values of n. The values of both the analytical and numerical areas are the same.

Comparing the plots of S against n, the plot for the analytical method is smoother than the numerical method for $n \leq 5$. This is expected as for the numerical integration method, the graph is plotting various values of S (evaluated at integer values of n) against n. For the analytical method, the area S is a function in n, thus the graph is plotting the smooth function. In addition, there are a greater number of plotted points on the analytical graph for $n \leq 5$, as opposed to the numerical graph. These factors explain why a smoother graph is obtained for the analytical solution.