Name: Adam von Arnim

# HW 1
# Draining Water Bottle

Math 466: Dynamic Modeling

Data was collected for a bottle draining out of a hole at its bottom.

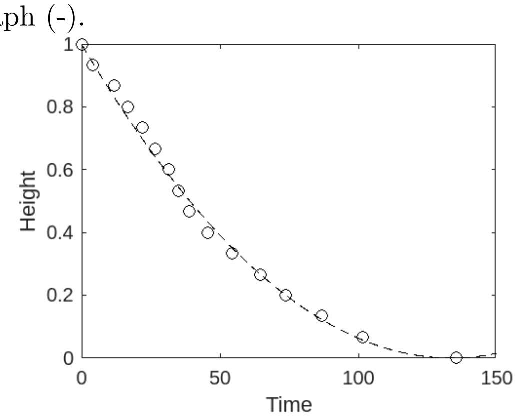
| height | time (sec) |
|---|---|
| 1 | 0 |
| 14/15 | 4.22 |
| 13/15 | 11.91 |
| 12/15 | 16.81 |
| ... | ... |
| 0 | 135.3 |

## 1

Find the value of k that best fits the data with model

$$h = (1 - kt)^2$$

To find the best k that fits this data, I created a numeric Matlab function and fit it using the `fitnlm` function. This fits a parameterized function based on the input data. The following graph was made using the input data (O) and the constructed graph (-).



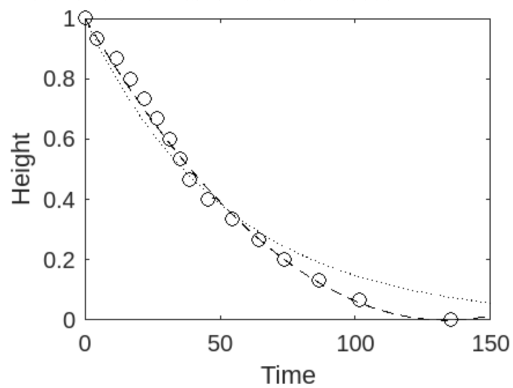

This indicated that 0.0074832 is the optimal value for k to satisfy the equation.

## 2

If we seek to fit our data to an exponential model, we can define a similar numeric function in Matlab and fit it similarly. We can plot these two lines side-by-side to get a qualitative view of how well they perform. In the graph below, we see the data (O), the first function (-), and the exponential function (.). Additionally, we can observe the error of the functions against the data to get a quantitative indicator of how well they perform.
The error of the initial function, given by

$$h = (1 - kt)^2$$

has a root mean squared error of 0.0232, whereas the exponential function has an error of 0.0532. Since the former is less than the latter, we know that it fits the data better.




## 3

Data was collected from a bottle with two holes in it: one at the midpoint, and one at the bottom of the bottle.

| height | time (sec) |
|---|---|
| 1 | 0 |
| 14/15 | 2.53 |
| 13/15 | 5.19 |
| 12/15 | 8.22 |
| ... | ... |
| 1/15 | 93.54 |
| 0 | 94.84 |

### 3a

We will modify the model above to find a model of how height changes. We assume that the second hole is of the same size as the first hole.
The initial 1-hole model asserts that there is some tie between potential energy from pressure at the hole and some kinetic energy as water flows out of it. Specifically,

$$KE \propto v^2$$

and

$$PE = Fd \propto h$$

. From the combination of these two,

$$KE \propto h$$

, and

$$\frac{dh}{dt} = -K\sqrt{h}$$

.
To adapt this model, we assume that there is pressure affecting the system at two places: point 1 and point 2. The midpoint hole should have less pressure, since its water depth is relative to the midpoint of the bottle

$$pressure_holeOne \propto (h - 0.5 * H)$$

. The pressure of the bottom hole should be consistent with the one-hole bottle

$$pressure_holeTwo \propto h$$

. Since the top hole is only effective while the overall height is greater than half of the total height, I propose using the following model:

$$\frac{dh}{dt} = \begin{cases} -\sqrt{A(h - 0.5 * H) + Ah}, & \text{if } h > H/2 \\ -\sqrt{Ah}, & \text{if } h \le H/2 \end{cases} \text{, where } A = (2k)^2$$

This simplifies to

$$\frac{dh}{dt} = \begin{cases} -\sqrt{2Ah - \frac{AH}{2}}, & \text{if } h > H/2 \\ -\sqrt{Ah}, & \text{if } h \le H/2 \end{cases}$$

### 3b

Using the value of k from (1), we can solve the differential equation by separation of variables. From (1), we found that a good value for k is 0.0074832.

$$When h > H/2, \frac{dh}{\sqrt{2Ah - AH/2}} = -dt,$$

yielding

$$-t + c = \frac{2\sqrt{2Ah - AH/2}}{2A} \implies t = \frac{-\sqrt{2Ah - AH/2}}{A} + c$$

Given (t=0, h=1), we solve for c:

$$c = \frac{\sqrt{2A - A(1)/2}}{A} = \frac{\sqrt{3A/2}}{A} = \sqrt{\frac{3}{2A}},$$

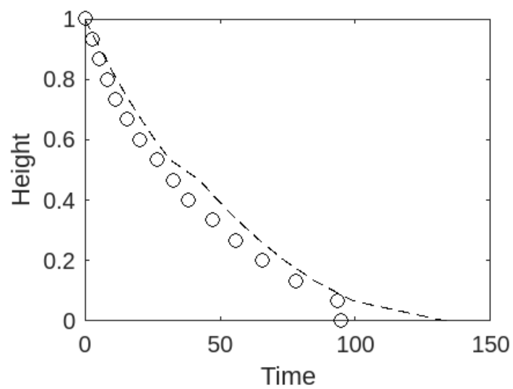
yielding

$$t(h) = \sqrt{\frac{3}{2A}} - \sqrt{2h/A - H/2A}, h > H/2$$

We then construct the full function:

$$t(h) = \begin{cases} \sqrt{\frac{3}{2A}} - \sqrt{\frac{4h-H}{2A}}, & \text{if } h > H/2 \\ \frac{\sqrt{h}-1}{k}, & \text{if } h \leq H/2 \end{cases}$$

Here we graph the data points and function approximating it, using the value for k found in problem 1: 0.0074832.

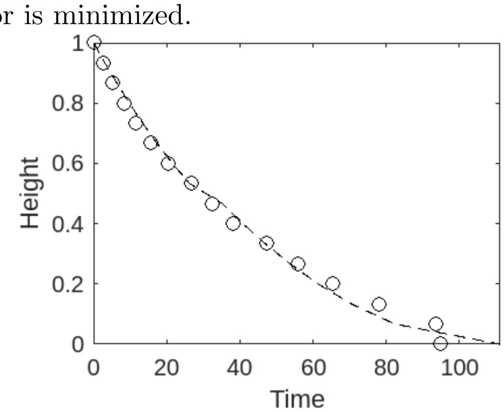


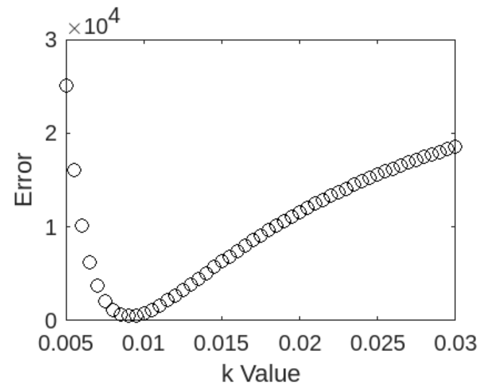

**3c**

By sampling a range of values k, we can find a better value for k (assuming that the value for the single hole in problem 1 does not apply as well here (with differently-sized and differently-placed holes).

I tested the range of [0.005, 0.03] at increments of 0.0005. The ideal value in this range and at this granularity was 0.0090. The graph of the best approximation is shown below. Notably, we see that its graph follows the data points much closer than the graph above especially in the 40-80 second time range.

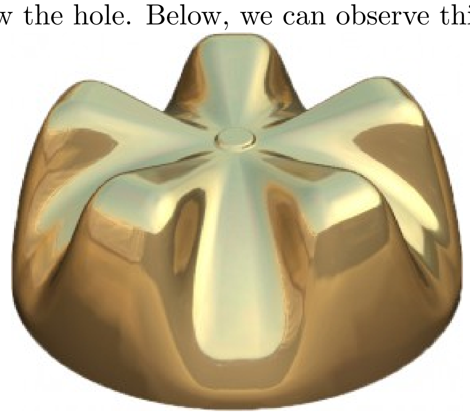
The graph of least-squares values is plotted below that. The ideal k value in the second graph is the point along the x-axis at which the error is minimized.





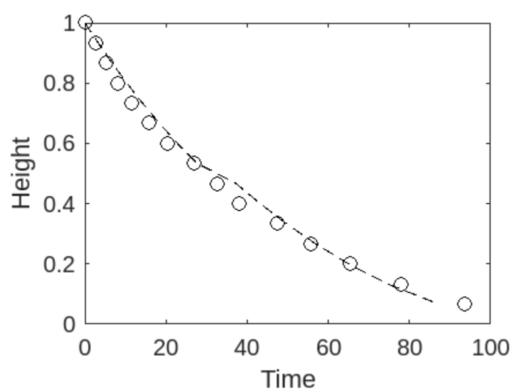

### 3d

Looking at these plots, one data point seems slightly off – the last data point, where a height of 0 is measured almost immediately after the height of 1/15. This might be the case because of it being pretty hard to tell when *exactly* there is no water left in the bottle (some drops may still exist or may have have collected in bottle ridges that are actually below the hole. Below, we can observe this sort of water bottle design.



Source: https://blog.krones.com/en/the-function-and-design-of-pet-container-bases/

If we assume that this data point is unreliable, we can find an improved k value (0.0085) and construct this revised graph:



# Code

Finding best k for 1 hole

```
xAxis= [0, 4.22, 11.91, 16.81, 21.95, 26.75, 31.44, 35.1, 38.8, 45.45, 54.36, 64.44, 73.78, 86.72, 101.51, 135.3];
yAxis  = 1:-1/15:0

plot(xAxis, yAxis, 'ko');
xlabel('Time');
ylabel('Height');

powerModelFunction = @(b,x) (1-b(1).*x).^2;

powerModel = fitnlm(xAxis, yAxis, powerModelFunction, [1])
xx = linspace(0,150)';
line(xx, predict(powerModel, xx), 'linestyle', '--', 'color', 'k')

expModelFunction = @(c,x) 1*(exp(c(1).*x));

exponentialModel = fitnlm(xAxis, yAxis, expModelFunction, [1])
line(xx, predict(exponentialModel, xx), 'linestyle', ':', 'color', 'k')
```

Finding predictions for 2 holes using initial k value

```
xAxis = [0, 2.53, 5.19, 8.22, 11.48, 15.69, 20.32, 26.82, 32.49, 38.14, 47.36, 55.8, 65.41, 77.99, 93.54, 94.84];
yAxis = 1:-1/15:0;

plot(xAxis, yAxis, 'ko');
xlabel('Time');
ylabel('Height');

A = 0.000223993129;
k = (A^0.5)/2;
H = 1;
timePredictions = zeros(length(yAxis),1);

for index=1:length(yAxis)
    height = yAxis(index);
    time = timePredictor(height, A, H, k);
    timePredictions(index) = time;
end

timePredictions
line(timePredictions, yAxis, 'linestyle', '--', 'color', 'k')


function [time] = timePredictor(height, A, H, k)
    if height > H/2
        time = (3/(2*A))^0.5 - (2*height/A - 1/(2*A))^0.5;
    else
        time = (1 - height^0.5) / k;
    end
end
```

Finding best k for two holes

```
xAxis = [0, 2.53, 5.19, 8.22, 11.48, 15.69, 20.32, 26.82, 32.49, 38.14, 47.36, 55.8, 65.41, 77.99, 93.54, 94.84];
yAxis = 1:-1/15:0;

plot(xAxis, yAxis, 'ko');
xlabel('Time');
ylabel('Height');

kOptions = 0.005:0.0005:0.03;
H=1;

minError = sum(xAxis.^2)
bestK = 0;
bestTimePredictions = zeros(length(xAxis), 1);

errors = zeros(length(xAxis), 1);

% iterate through options for k to find minimum error
for index=1:length(kOptions)
    k = kOptions(index);
    A = (2*k)^2;
    timePredictions = zeros(length(yAxis), 1);

%     get prediction values
    for dataPoint=1:length(yAxis)
        height = yAxis(dataPoint);
        time = timePredictor(height, A, H, k);
        timePredictions(dataPoint) = time;
    end

%     calculate error

    error = sum((transpose(timePredictions)-xAxis).^2)
    errors(index) = error
    if minError > error
        bestK = k;
        bestTimePredictions = timePredictions;
        minError = error;
    end
end

bestK
```

```
line(bestTimePredictions, yAxis, 'linestyle', '--', 'color', 'k')

% opens new figure
figure()
plot(kOptions, errors, 'ko');
xlabel('k Value');
ylabel('Error');


function [time] = timePredictor(height, A, H, k)
    if height > H/2
        time = (3/(2*A))^0.5 - (2*height/A - 1/(2*A))^0.5;
    else
        time = (1 - height^0.5) / k;
    end
end
```