

Introduction to the Finite Difference  
Method:  
Filling and Draining a Cylindrical Tube

Lensyl Urbano

Monday 31<sup>st</sup> October, 2022



# Chapter 1

## Filling a Cylindrical Tube

Consider filling a cylinder with water.

The water flows in at a constant rate of  $5 \text{ cm}^3/\text{s}$ . The inflow rate ( $Q$ ) can be written as the change in volume ( $V$ ) over the change in time ( $t$ ) (the  $\Delta$  symbol represents change):

**Inflow Rate**

$$Q = \frac{\Delta V}{\Delta t} = 5 \text{ cm}^3/\text{s} \quad (1.1)$$

### 1.1 Filling the Tube Calculations and Equations

#### 1.1.1 Conceptual Physics Approach

So, at this inflow rate, after 10 seconds there will be  $50 \text{ cm}^3$  added to the cylinder.

$$\begin{aligned} V &= 5 \text{ cm}^3/\text{s} \cdot 10 \text{ s} \\ &= 50 \text{ cm}^3 \end{aligned}$$

In terms of the equation, the change in volume is equal to the inflow rate ( $Q$ ) times the time period ( $t$ ).

$$V = Q \cdot t \quad (1.2)$$

How high will the water have risen in the cylinder in those 10 seconds when  $50 \text{ cm}^3$  of water was added? Well, we know that the

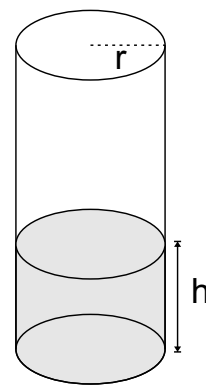


Figure 1.1: Cylinder with dimensions.  $r$  is the radius and  $h$  is the height of water in the tube.

volume of a cylinder is given by the equation:

$$V = \pi r^2 h \quad (1.3)$$

So, if we know the volume and the radius of the cylinder ( $r$ ) we can solve this equation for the height ( $h$ ):

Divide both sides by  $\pi r^2$ :

$$\frac{V}{\pi r^2} = \frac{\pi r^2 h}{\pi r^2} \quad (1.4)$$

To get:

$$\frac{V}{\pi r^2} = h \quad (1.5)$$

Which can be rewritten as:

$$h = \frac{V}{\pi r^2} \quad (1.6)$$

or:

$$h = \frac{1}{\pi r^2} V \quad (1.7)$$

Thus, for our given problem where the radius is 2.25 cm, and the volume of water added is 50 cm<sup>3</sup>:

$$h = \frac{1}{\pi \cdot 2.25^2} \cdot 50 = 3.1 \text{ cm} \quad (1.8)$$

Now we can substitute for volume using Equation 1.2 to get:

$$h = \frac{1}{\pi r^2} Q \cdot t \quad (1.9)$$

Now, lets rewrite this equation so we just consider what happens over a small time period (call it a *time step* denoted by  $\Delta t$ ). It's the change from moment to moment and results in a small change in height ( $\Delta h$ ). So our final equation becomes:

$$\Delta h = \frac{1}{\pi r^2} Q \cdot \Delta t \quad (1.10)$$

Which we rearrange a little to get:

$$\boxed{\Delta h = \frac{\Delta t}{\pi r^2} Q} \quad (1.11)$$

Having calculated the change in the height of the water in the cylinder in a given time step ( $\Delta t$ ), for each timestep we calculate the new height of water ( $h_{new}$ ) as the old height plus the change:

$$\boxed{h_{new} = h_{old} + \Delta h} \quad (1.12)$$

We can now use these two equation to create a computer model that gives the height of water in the column over time.

### 1.1.2 Using Calculus to Find the Discrete Equations

Same problem—filling a cylinder—but using calculus to end up with the same equations in the end.

Start with the equation for the volume of a cylinder:

$$V = \pi r^2 h \quad (1.13)$$

There are two variables that change with time as the cylinder fills, the volume ( $V$ ) and the height ( $h$ ) since the radius ( $r$ ) does not change. So, if we differentiate this equation with respect to time (implicit differentiation), we get:

$$\frac{dV}{dt} = \pi r^2 \frac{dh}{dt} \quad (1.14)$$

Solving for  $\frac{dh}{dt}$  gives the **height change equation**:

$$\frac{dh}{dt} = \frac{1}{\pi r^2} \frac{dV}{dt} \quad (1.15)$$

The expression  $\frac{dh}{dt}$  represents the instantaneous change in height with time: the rate at which height changes at any instant. To write a program to solve this equation we'll **discretize** the expression by using  $\frac{\Delta h}{\Delta t}$ :

$$\frac{\Delta h}{\Delta t} \approx \frac{dh}{dt} \quad (1.16)$$

The  $\Delta$  means that we're taking the difference between two discrete value of  $h$ , so:

$$\Delta h = h_2 - h_1 \quad (1.17)$$

Since this is the rate of change over time it can be easier to think of the change in height as the difference between the new height and the old height over the short ( $\Delta t$ ) time period.

$$\Delta h = h_{new} - h_{old} \quad (1.18)$$

So now we rewrite our height change equation (Eqn. 1.15) as:

$$\frac{\Delta h}{\Delta t} = \frac{1}{\pi r^2} \frac{dV}{dt} \quad (1.19)$$

which we can solve for the change in height ( $\Delta h$ ):

$$\Delta h = \frac{\Delta t}{\pi r^2} \frac{dV}{dt} \quad (1.20)$$

Since the inflow rate ( $Q$ ) is the change in volume over time, and it remains constant for our model, we can say:

### Change in Height Equation

$$\boxed{\Delta h = \frac{\Delta t}{\pi r^2} Q} \quad (1.21)$$

Which is the same equation (Eqn. 1.11) we found when we took the conceptual approach in the previous section.

Now we substitute in the discrete change for  $\Delta h$  (Eqn. 1.18) to get:

$$h_{new} - h_{old} = \frac{\Delta t}{\pi r^2} Q \quad (1.22)$$

Which we can solve for the new height:

$$h_{new} = h_{old} + \frac{\Delta t}{\pi r^2} Q \quad (1.23)$$

Which is the same as:

### Height Update Equation

$$\boxed{h_{new} = h_{old} + \Delta h} \quad (1.24)$$

We can use the **Change in Height** (Eqn. 1.21) and **Height Update** (Eqn. 1.18) equations to create a computer model of the height of the water in the cylinder as it fills it up.

## 1.2 Code

The following example code that solves this water-filling problem uses the ezGraph class (<https://github.com/lurbano/ezGraph>) which requires matplotlib and numpy. However, the code in the following section avoids the use of most imported modules, but does not graph.

### Code with Graphical Output

*water-filling-fd.py*

```

1  import numpy as np
2  import time
3  from ezGraph import *
4
5  # Finite Difference Model
6
7  # PARAMETERS
8  dt = 1.
9  nsteps = 20
10
11  r = 2.25      # radius (cm)
12  Qin = 5      # Volume inflow rate: (cubic cm / s
13              # )
14  h = 0        # Initial height (cm)
15
16  # GRAPH
17  graph = ezGraph(xmax=30, ymin=0, ymax=10,
18                  xlabel="Time_(s)", ylabel="Height_(cm)")
19  graph.add(0, h)      # add initial
20                       # values
21
22  # TIME LOOP
23  for t in range(1, nsteps):
24      modelTime = t * dt
25
26      dh = Qin * dt / (np.pi * r**2)      # find
27                                             # the change in height
28      h = h + dh                            # update
29                                             # height
30
31      print(modelTime, h)
32      graph.add(modelTime, h)

```

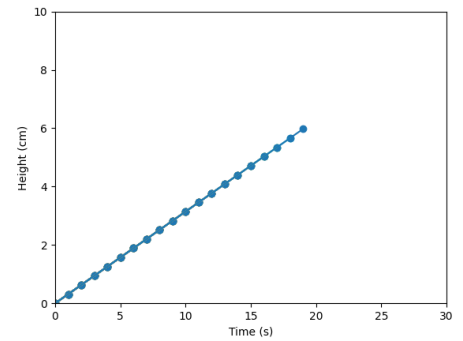


Figure 1.2: Model output: Graph of height of water in the column over time when filling the cylinder.

```

29     graph.wait(0.1)
30
31 # DRAW GRAPH
32 graph.keepOpen()

```

### 1.2.1 Code without Graphical Output

A stripped down version of the code with no graph and no external modules except "math".

*water-filling-fd-noGraph.py*

```

1  import math
2  # Finite Difference Model
3
4  # PARAMETERS
5  dt = 1.
6  nsteps = 20
7
8  r = 2.25      # radius (cm)
9  Q = 5         # Volume inflow rate: (cubic cm / s
10 )
11 h = 0         # Initial height (cm)
12
13 print(0, h)    # print initial values
14
15 # TIME LOOP
16 for t in range(1, nsteps):
17     modelTime = t * dt
18
19     dh = Q * dt / (math.pi * r**2)    # find
20         the change in height
21     h = h + dh                          # update
22         height
23
24     print(modelTime, h)

```

Which should produce a table of time and height output:

```

1  0 0
2  1.0 0.31438013450250935
3  2.0 0.6287602690050187
4  3.0 0.943140403507528
5  4.0 1.2575205380100374
6  5.0 1.5719006725125468

```



7	6.0	1.8862808070150563
8	7.0	2.2006609415175657
9	8.0	2.515041076020075
10	9.0	2.8294212105225847
11	10.0	3.143801345025094
12	11.0	3.4581814795276036
13	12.0	3.772561614030113
14	13.0	4.086941748532622
15	14.0	4.4013218830351315
16	15.0	4.715702017537641
17	16.0	5.03008215204015
18	17.0	5.34446228654266
19	18.0	5.658842421045169
20	19.0	5.973222555547679

### 1.3 Analytical Solutions using Calculus

The analytical solution to this problem will help confirm the accuracy of our model. +

#### 1.3.1 Filling

As we saw in the section on using calculus (Section 1.1.2), we can start with the equation for the volume of a cylinder:

$$V = \pi r^2 h \quad (1.25)$$

And differentiate with respect to time to get:

$$\frac{dV}{dt} = \pi r^2 \frac{dh}{dt} \quad (1.26)$$

Assuming a constant inflow rate ( $\frac{dV}{dt} = Q$ ):

$$Q = \pi r^2 \frac{dh}{dt} \quad (1.27)$$

And solve for  $\frac{dh}{dt}$ :

$$\frac{dh}{dt} = \frac{Q}{\pi r^2} \quad (1.28)$$

This we can separate:

$$dh = \frac{Q}{\pi r^2} dt \quad (1.29)$$

and integrate:

$$\int dh = \int \frac{Q}{\pi r^2} dt \quad (1.30)$$

to get:

$$h = \frac{Q}{\pi r^2} t + c \quad (1.31)$$

When  $t = 0$ ,  $c$  can be shown to be the initial height ( $c = h_i$ ) so:

$$h = \frac{Q}{\pi r^2} t + h_i \quad (1.32)$$

And since  $\frac{Q}{\pi r^2}$  is constant, we can see that this term is the slope in a linear equation of the form.

$$y = mx + b \quad (1.33)$$

So the linear pattern produced by the filling model is correct (Figure 1.2).

## Chapter 2

# Draining

Consider a cylinder with water draining out of the bottom through a hole.

### 2.1 Numerical Solution

For draining, the outflow rate (change in volume over time, ( $Q = \frac{dV}{dt}$ ) is not constant. The outflow rate is proportional to the height of water in the tube, since the higher the water level the greater pressure at the bottom of the tube and the faster the outflow rate.

$$\frac{dV}{dt} \propto h \quad (2.1)$$

Converting the proportionality statement to an equation requires us to introduce a constant ( $k$ ). Also, recognizing that this will be an outflow rate means that the flow rate should be negative:

$$\frac{dV}{dt} = -k \cdot h \quad (2.2)$$

As we saw when we were filling the cylinder, the change in height of water in the tube is given by the **change in height** equation:

$$\Delta h = \frac{\Delta t}{\pi r^2} Q \quad (2.3)$$

where  $Q$  is  $\frac{dV}{dt}$  so:

$$\Delta h = \frac{\Delta t}{\pi r^2} \frac{dV}{dt} \quad (2.4)$$

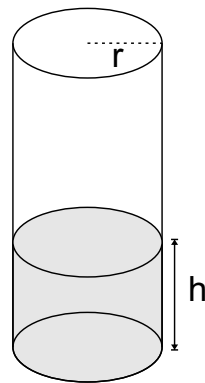


Figure 2.1: Cylinder with dimensions.  $r$  is the radius and  $h$  is the height of water in the tube.

So, let's substitute our flow rate equation (Eqn. 2.2) for  $\frac{dV}{dt}$  to get:

$$\Delta h = \frac{\Delta t}{\pi r^2} (-k \cdot h) \quad (2.5)$$

which simplifies to:

$$\Delta h = -\frac{\Delta t}{\pi r^2} k \cdot h \quad (2.6)$$

Important to note for the computer model, is that the height ( $h$ ) used in this equation is the old height from the previous timestep so:

$$\Delta h = -\frac{\Delta t}{\pi r^2} k \cdot h_{old} \quad (2.7)$$

and we can still update the new height using:

$$h_{new} = h_{old} + \Delta h \quad (2.8)$$

so, in our code we just need to change this line and set up a few different constants.

### 2.1.1 Code: Draining

This program is based off the filling code, but for this example we ignore the filling by setting the inflow rate to zero (**Line 12**). We're using an initial height of 50 cm ( $h_0 = 50$ ), and set the constant  $k$  to be equal to one.

*water-draining-fd.py*

```

1 import numpy as np
2 import time
3 from ezGraph import *
4
5 # Finite Difference Model
6
7 # PARAMETERS
8 dt = 1
9 nsteps = 100
10
11 r = 2.25      # radius (cm)
12 Qin = 5      # Volume inflow rate (dV/dt): (
    cubic cm / s)
```

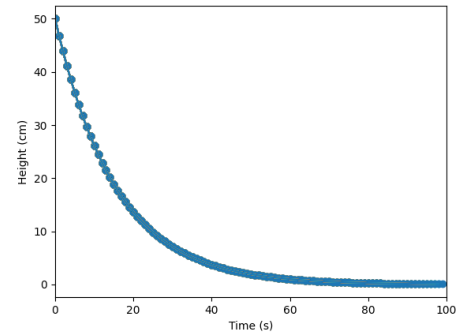


Figure 2.2: Model output: Graph of height of water in the column over time when draining the cylinder via a hole in the bottom.

```

13 h = 0          # Initial height (cm)
14 k = 1.0        # outflow rate constant
15
16 # GRAPH
17 graph = ezGraph(xmax=100,
18                 xlabel="Time_(s)", ylabel="
19                 Height_(cm)")
20 graph.add(0, h) # add initial
21                 values
22
23 # TIME LOOP
24 for t in range(1, nsteps):
25     modelTime = t * dt
26
27     # Filling
28     dh = Qin * dt / (np.pi * r**2) # find the
29                                     change in height
30     h = h + dh                      # update
31                                     height
32
33     # Draining
34     dVdt = -k * h
35     dh = dVdt * dt / (np.pi * r**2)
36     h = h + dh
37
38     print(modelTime, h)
39     graph.add(modelTime, h)
40     graph.wait(0.1)
41
42 # DRAW GRAPH
43 graph.keepOpen()

```

The output from the model (Fig. 4.1) looks like an exponential decay curve, which is what we will find from the analytical solution (Eqn. 2.24).

## 2.2 Draining: Analytical Solution using Calculus

Experiments (which you may have done) show that if you're draining a cylinder by gravity the outflow rate of water is linearly pro-

portional to the height of water in the tube.

$$\frac{dV}{dt} \propto h \quad (2.9)$$

Converting the proportionality statement to an equation requires us to introduce a constant ( $k$ ):

$$\frac{dV}{dt} = kh \quad (2.10)$$

So in draining, the outflow rate ( $\frac{dV}{dt}$ ) is not constant, it slows down as the height of water in the tube decreases.

Now, lets substitute the equation for the volume of a cylinder:

$$V = \pi r^2 h \quad (2.11)$$

into the draining equation (Eq. 2.10) to get:

$$\frac{d[\pi r^2 h]}{dt} = kh \quad (2.12)$$

we can extract  $\pi$  and  $r^2$  from the differential because they are constant:

$$\pi r^2 \frac{dh}{dt} = kh \quad (2.13)$$

separating the variables gives:

$$\pi r^2 \frac{dh}{h} = k \cdot dt \quad (2.14)$$

and rearranging:

$$\frac{dh}{h} = \frac{k \cdot dt}{\pi r^2} \quad (2.15)$$

$$\frac{dh}{h} = \frac{k}{\pi r^2} dt \quad (2.16)$$

To simplify a little, lets consolidate the constants on the left hand side into one variable  $K$ :

$$K = \frac{k}{\pi r^2} \quad (2.17)$$

so:

$$\frac{dh}{h} = K \cdot dt \quad (2.18)$$

which we can integrate (remember  $K$  is a constant):

$$\int \frac{dh}{h} = K \int dt \quad (2.19)$$

$$\ln h = K \cdot t + c \quad (2.20)$$

we can solve for  $h$  by raising both sides by  $e$  to cancel the  $\ln$ :

$$e^{\ln h} = e^{Kt+c} \quad (2.21)$$

$$h = e^{Kt+c} \quad (2.22)$$

Because of math, we can pull the constant out to get:

$$h = Ce^{Kt} \quad (2.23)$$

Where the constant is the initial value of the height ( $h_0$ ):

$$h = h_0 \cdot e^{Kt} \quad (2.24)$$

This is an exponential function. If  $K$  is less than 1 ( $K < 1$ ) then this is a decay curve.





# Chapter 3

## Model Calibration

We have a model that shows the general patterns we expect when filling and draining the tube: linear increase for filling at a constant rate, and exponential decay for draining due to gravity. But can these models reflect the actual thing?

Fortunately, we have some experimental data, thanks to my pre-Calculus class.

We'll start with the filling portion of the model.

### 3.1 Filling Calibration

time (s)	height (cm)
1	0
7	10
12	20
16.8	30
21.5	40
26.2	50

Table 3.1: Experimental Data from filling the cylinder. Data by Blas.

Recall that, at the core of the model, is the change in height equation (Eqn. 1.11):

$$\Delta h = \frac{\Delta t}{\pi r^2} Q$$

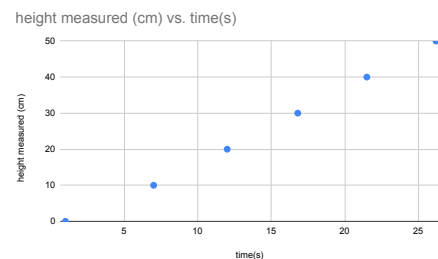


Figure 3.1: Experimental data from filling the cylinder. Data by Blas.

The radius ( $r$ ) is measured and we choose  $\Delta t$  as steps in the simulation, so the only unknown is the inflow rate ( $Q$ ).

Therefore, to calibrate the model we adjust  $Q$  until the model output matches the experimental data.

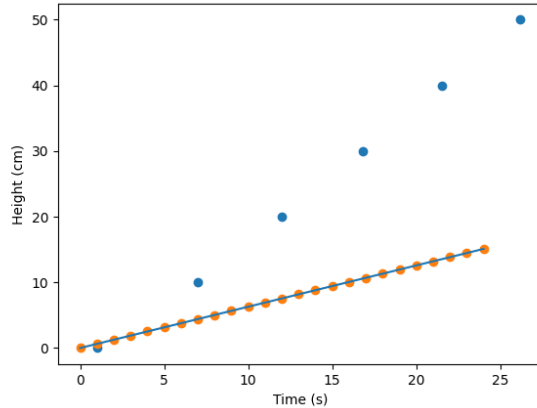


Figure 3.2: Comparison of measured (blue) and modeled (red) data. This version of the model uses  $Q = 10 \text{ cm}^3/\text{s}$ .

In order to produce this graph, we use the *ezGraphMM* class, which allows us to plot the measured and modeled data separately. The full code is here:

*water-filling-calibration.py*

```

1 import numpy as np
2 import time
3 from ezGraph import *
4
5 # Finite Difference Model
6
7 # PARAMETERS
8 dt = 1
9 nsteps = 30
10
11 r = 2.25      # radius (cm)
12 Qin = 30      # Volume inflow rate (dV/dt): (
13               # cubic cm / s)
14 h = 0         # Initial height (cm)
15 k = 0.0       # outflow rate constant
16
17 # EXPERIMENTAL DATA

```

```

17 x_measured = [1, 7, 12, 17, 22, 26]
18 y_measured = [0, 10, 20, 30, 40, 50]
19 y_modeled = []
20
21 # GRAPH
22 graph = ezGraphMM(xmin=0, xmax=100,
23                   xLabel="Time_(s)",
24                   yLabel="Height_(cm)",
25                   x_measured = x_measured,
26                   y_measured = y_measured)
27
28 graph.addModeled(0, h)          # add
    initial values
29
30
31 # TIME LOOP
32 for t in range(1, nsteps):
33     modelTime = t * dt
34
35     # Filling
36     dh = Qin * dt / (np.pi * r**2) # find the
        change in height
37     h = h + dh                    # update
        height
38
39     # Draining
40     dVdt = -k * h
41     dh = dVdt * dt / (np.pi * r**2)
42     h = h + dh
43
44     if (modelTime in x_measured):
45         print(modelTime, h)
46         y_modeled.append(h)
47         graph.addModeled(modelTime, h)
48         graph.wait(0.1)
49
50 print("h_measured:", y_measured)
51 print("h_modeled:", y_modeled)
52
53 # calculate average values for y_measured and
    y_modeled
54
55 # DRAW GRAPH
56 graph.keepOpen()

```

### 3.2 Draining Calibration

time (s)	height (cm)
0	50
28	40
60	30
101	20
157	10

Table 3.2: Experimental Data from draining the cylinder. Data by Coen and Blas.

**Assignment:** Use this experimental data to calibrate the draining portion of the model.

### 3.3 Breaking the Model

Experiment by changing the parameters to find the limits of the model. Questions to ask are:

1. What is the largest timestep you can use with this model.
2. What happens when you change the  $k$  coefficient? How large can you make it?

What you should find is that at for some values the model starts to oscillate out of control. This is one of the limitations of finite difference models. It requires pretty extreme settings in this example, but other models the unstable parameter values can be much closer to the values we want to use.

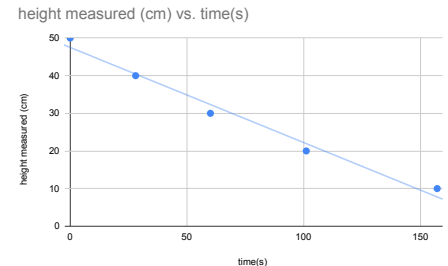


Figure 3.3: Experimental data from draining the cylinder. A linear trendline is included to highlight the curvature of the data. Data produced by Coen and Blas.

## Chapter 4

# Expanding the Model

As we've seen, modifying the basic model for different conditions can be pretty easy. In the base model (water filling) the change in height equation (Eqn. 1.11) was:

$$\Delta h = \frac{\Delta t}{\pi r^2} Q$$

To adjust for draining (Eqn. 2.7), we just needed to make the inflow value ( $Q$ ) an equation that adjusts for the fact that the water outflow rate is negative and depends on the height of water in the tube.

$$\Delta h = \frac{\Delta t}{\pi r^2} (-k \cdot h)$$

### 4.1 Simultaneously Filling and Draining

Our program is already set up to handle filling and draining. We simply need to change the inflow value ( $Q_{in}$ ).

**What happens if we try to fill the tube while the hole in the bottom is open so it simultaneously drains?**

Set the inflow value to 5 cm<sup>3</sup>/s, set the initial height ( $h$ ) to 0, and keep the draining parameters, we should see the water level increase until it reaches an equilibrium.

**Model Equilibrium:** Now, if you have not already, set the inflow value and  $k$  coefficient to the values you got from your calibrations and determine the equilibrium height of water in the tube.

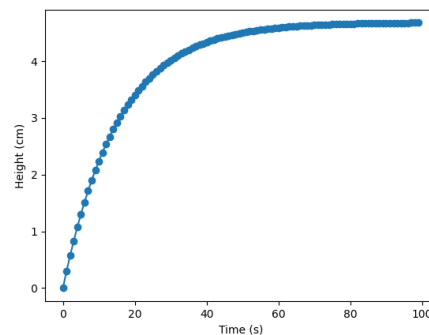


Figure 4.1: The model indicates that filling the cylinder while the hole is still open would result in the water rising to an equilibrium height of about 4.7 cm.



# Chapter 5

## Exercises

### 5.1 Rectangular Prism

Create a model for the filling of a water container where the container was a rectangular prism instead of a cylinder, where the radius increases by 1 cm for every 2 cm increase in height?

### 5.2 Cone

Create a model for the filling of a water container where the container was conical instead of a cylinder, where the radius increases by 1 cm for every 2 cm increase in height?

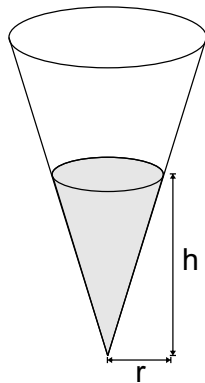


Figure 5.1: Conical water container.