# Notes on Overflowing

The exploration began with the following task, question #25 of the 2002 Pascal contest run by the Centre for Education in Mathematics and Computing at the University of Waterloo, quoted in Godin (2013).

A student has two open-topped cylindrical containers. The larger container has a height of 20 cm, a radius of 6 cm and contains water to a depth of 17 cm. The smaller container has a height of 18 cm, a radius of 5 cm and is empty. The student slowly lowers the smaller container into the larger container. As the smaller container is lowered, the water first overflows out of the larger container and then eventually pours into the smaller container. Determine the depth of the water in the smaller container when the smaller container is resting on the bottom of the larger container.

My interest was in developing an animated applet in which the sizes of the bucket and the can may be altered, as well as the amount of water initially in the bucket.

I started with the animation, but as I struggled to get all the conditions correct, I moved to drawing a graph of the water level, with intermediate calculations also displayed so as to be able to check. Then I had to revise the animations based on the graphing results.

## Possible Tasks

The original task.

What relationships must there be between the initial, bucket and can volumes so that the can ends up full when it gets to the bottom (and the water level is at the top of the can)?

Predict the water height as a function of the distance of the bottom of the can from the top of the bucket, in terms of the can and bucket sizes, and the initial volume of water.

Think in terms of ratios rather than differences.

## Using the App

White points can be varied. The base of the can may be raised and lowered by dragging the mid-point.

The widths of the bases of the buckets and can may be adjusted by dragging, as too the heights of the sides (on the left) and the initial water level.

#### Change Button

Clicking changes from a 2D to a 3D model. First it assumes the bucket and the can as cuboids, then that they are circular cylinders.

#### Equal Button

When pressed, the outside spill tray has the same cross section as the can.

#### Ratios Table Button

This reveals relevant ratios for thinking in ratio terms

#### Current Values Button

Displays current values of water level in bucket, level beside (from bottom) can and inside can (from bottom of can), and above bucket

#### Probe Values Button

Allows probing of values according to value of bottom of can below top of bucket without disturbing the picture.

#### Graphs Button

Displays graphs of water level in bucket, beside and inside can

#### Labels Button

Labels the graphs

## Pedagogical Choices

Having imagined the situation without specific measurements, it quickly becomes evident that there are various possible structural relationships in order to decide what the water level graph might look like.

It might be tempting to begin with the bucket originally filled, before considering the more general situation.

There are opportunities to interpret the graph, imagining the can-bucket-water situation at various critical points where the graph changes. That is why the graphs can be displayed unlabelled, so that they can be interpreted. Alternatively the probe and current values can be used to imagine the graphs first.

The ratios table can be used to set tasks based on relative values rather than on absolute values or differences

# Reference

Godin, S. (2013). What’s the problem? Cylinders within cylinders. *Ontario Mathematics Gazette*. 51(4), 8-9.

# One version of water level function

Water Level(t) :=(

SpillVol = max( 0, min (t, Can Height)\*Can Base+VolWater – Bucket Vol);// vol spilled outside of bucket

H1 = max(0, Bucket Height – t]); *//*ht of can bottom above bucket bottom

H2 = min( Initial Height in Bucket, H1); *//*ht of water below can

V2 = H2\*LBase; *//*vol of water below can

V3A = VolWater – V2 – SpillVol; *//*Vol of water not below can

H3A = V3A/(Bucket Base – Can Base); *//*theoretical height of water not below can

H3 = min(H3A, Can Height); *//*height of water beside can outside it

V3 = H3 (Bucket Base – Can Base); *//*Vol water beside can

V3B = H3\*Bucket Base; *//*Vol of water beside can plus can displacement

HSpill = SpillVol/(SBase); *//*equivalent height of spill in terms of can

HSpillA = SpillVol/(LBase-SBase); *//*equivalent height of spill outside can

H4A=max(0, H3A-HS); *//*theoretical height of water above can as if beside can

H4B = min(H4A, Can Height]); *//*height of water beside can

V4 = H4B\*(Bucket Base – Can Base);

H4 = min(V4/Can Base, Can Height]); *//*Height of water inside can

V5A = H4A-H4B;

H5=max( 0, V3A-V3-H4\*Can Base/Bucket Base; *//* height of water above can top

H2 + H3 + H5;