

# Saving the Crew of a Sinking Ship (Ship Outta Luck)

*Irene Li (mengzeli) and Jordan Tick (jrtick)*

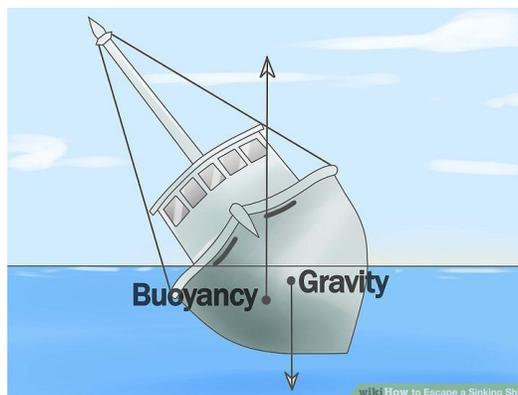
May 9th, 2017

## Abstract

The goal of our project is to model a sinking ship with the intent of proving that rescue vehicles sent at regular speeds can save all passengers within a certain time frame. To achieve this, we broke our project down into two models and then combined them to form a third. In model 1, we prove a lower bound on the time it takes for a ship (modeled as an upright prism) to sink. In model 2, we prove a lower bound on the amount of time needed saved all passengers on the ship. Finally, model 3 combines our previous models to prove that certain characteristics of our system determine how much time is needed to save all passengers.

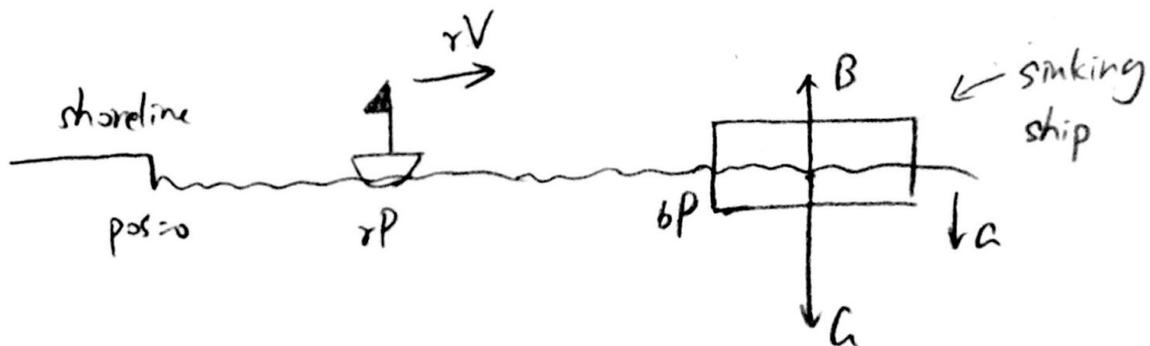
## 1 Introduction

Ship sinking dynamics are complex. At the very core of this, however, there are two forces driving the sink rate: gravity and buoyancy. The difficulty lies in the fact that these forces are constantly changing in such a system: gravity is increasing as water enters the ship (causing an increase in its weight) whereas buoyancy is increasing as the ship is dragged down further, displacing more and more water.



The intermingling of buoyancy and gravitational forces inherently creates an interesting differential equation which we attempt to make a lower bound on. By proving this bound, we show the relations of various aspect of nature as it relates to the sinking of a ship.

In our model itself, as shown below, we have a fleet of rescue vehicles moving between the sinking ship and the shore to save passengers. If it is at the ship, it picks up passengers; if it is at the shore, it drops off the passengers.



## 2. Brief Explanation of Related Terms

In proving our model, we used a variety of techniques. A short index of these terms is given below for those unaware of Dynamic Logic tactics.

**Loop:** a sequence of steps repeated for any number of iterations.

**Loop invariant:** an invariant that holds true throughout a loop.

**Differential invariant (DI):** an invariant that holds true throughout continuous dynamics described by differential equations.

**Differential ghost:** An additional auxiliary variable introduced for the sake of the proof.

**Differential cut rule (DC):** In order to prove property A of differential equations, this rule cuts in some other property B and splits the proof into two subgoals: one is to prove B as a differential invariant; the other is to prove A with the assumption that B holds true throughout.

**QE:** A tactic of proving an equation holds true by basic mathematics.

### 3 Explanation and Justification of Approach

#### 3.1 Model 1: ship rate model

To model gravitational and buoyancy forces, our core equations are these:

$$F_{gravity} = ((mass\ of\ ship) + (volume\ of\ water\ on\ ship) * (water\ density)) * (gravitational\ constant)$$

$$\text{Where } volume\ of\ water\ on\ ship = (flow\ of\ water\ onto\ ship) * time$$

$$F_{buoyancy} = (volume\ of\ ship\ underwater) * (water\ density) * (gravitational\ constant)$$

$$\text{Where } volume\ of\ ship\ underwater = (ship\ depth) * (ship\ width) * (ship\ length)$$

$$F_{external} = F_{gravity} + F_{buoyancy}$$

$$(ship\ depth)''(t) = F_{external} / (mass\ of\ ship)$$

Assumptions and simplifications:

1. Here, the sinking acceleration (i.e, (ship depth)''(t)) is not completely accurate. Since water is flowing into the ship, the ship mass is constantly changing. Thus, by differentiating the equation  $P = m*v$  (momentum is the product of mass and velocity), we find that  $dP/dt = dm/dt * v + m * dv/dt$ , where  $dP/dt$  is the external force and  $dv/dt$  is the acceleration. So the accurate acceleration would be:

$$(F_{external} - v * dm/dt)/m(t)$$

However, using the accurate acceleration would make the differential equation far more complicated and almost impossible to solve. After further discussion, we agreed that it is not necessary to use the accurate acceleration because we are modeling a "slowly" sinking ship (if the ship sinks in a few minutes, it would be impractical to talk about sending out rescue vehicles to save anyone from the ship) and therefore the rate of water flowing into the ship is fairly low, i.e,  $dm/dt$  is small. Thus, we decide it is reasonable to omit the  $-v*dm/dt$  term in the expression for acceleration. Now our acceleration becomes  $(F_{external})/m(t)$ . We further simplified the expression by assuming that  $m(t)$  remains constant. While this is not true since water is flooding into the ship, we made this assumption considering that the total mass of water getting into the ship is probably small compared to the mass of the ship itself (In real world, ships that just sink down

often have lots of air chambers in it and are not completely filled with water). With these two simplifications, we have our final expression for acceleration:

$$(\text{ship depth})''(t) = F_{\text{external}} / (\text{mass of ship})$$

2. We assumed the ship doesn't flip or tilt during the sinking process. The only motion involved is in the vertical direction.

**For simplicity within this paper, we will use many symbols to represent our system constants. A comprehensive list of our variables are as follows:**

*time since start of simulation = t*

*mass of ship = m*

*water density = p*

*| depth of ship | = d*

*rate at which depth is increasing = dV*

*height of ship = h*

*Cross section of ship = w \* l = A*

*| gravitational constant | = g*

*amount of time it takes for ship to sink = T*

*flow of water into ship (m<sup>3</sup>/s) = hole*

*volume of water on ship = waterOnBoard*

Now, with these equations in mind, we can write out our core equations again in symbolic form:

$$F_{gravity} = (m + waterOnBoard * p) * g$$

$$waterOnBoard = hole * t$$

$$F_{buoyancy} = (d * A) * p * g$$

$$d''(t) = dV'(t) = (m + hole * t * p - d * A * p) * g / m$$

There are two main aspects of note for our system:

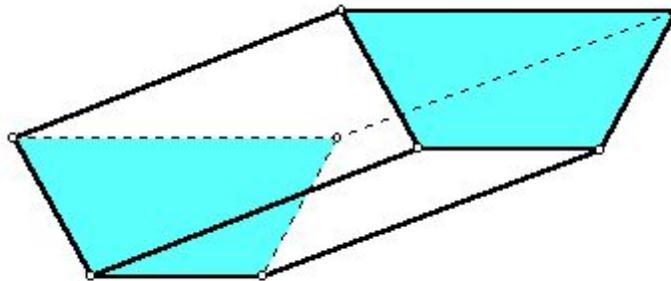
First, **we model ship depth as the positive distance that the ship is underwater**; as such, gravity is modelled as a positive magnitude force whereas buoyancy has a negative magnitude as it is attempting to push the ship up to the surface.

Second, our system models the inhomogeneous second order ordinary differential equation

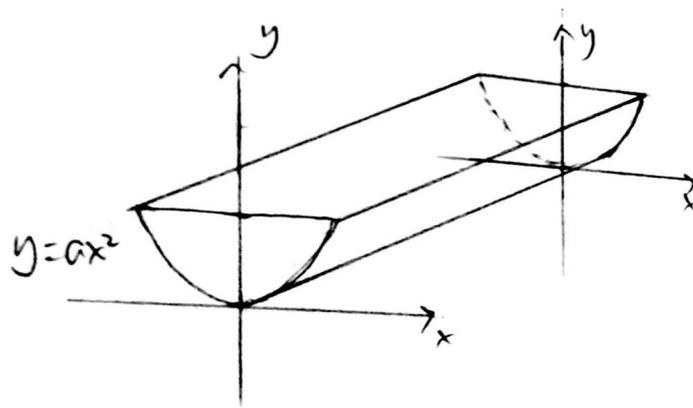
$$d''(t) + (Apg/m) * d(t) = (hole * pg/m) * t + g$$

The majority of our time spent on this model was in solving this differential equation and giving a sensible bound that KeYmaeraX would be able to prove. The difficulty of solving such an equation is directly related to the model of ship that we use, which is why we decided to use an upright prism as our ship model. For reference, here are the equations of other three-dimensional shapes we considered (where c is a constant):

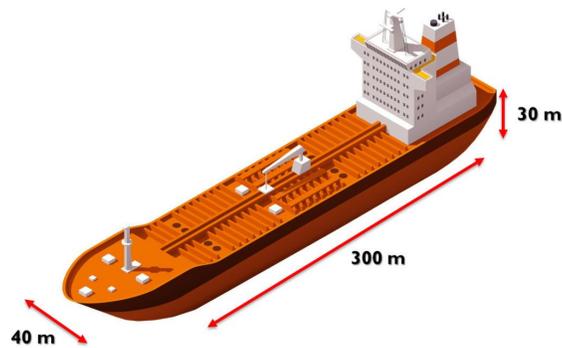
Sideways Trapezoidal Prism:  $d''(t) + pg/m * d(t) * (d(t) + c) = (hole * pg/m) * t + g$



Sideways Parabolic Prism:  $d''(t) + pg/m * c * d(t)^{3/2} = (hole * pg/m) * t + g$



Cones and ellipsoids were also considered but proved to be equally difficult (if not more so). Because we wanted to make sure our system was solvable with a sensible bound, we chose to stick with a prism (with base facing downward). In theory, many ships approximately adhere to this model, as this ship shown below can be roughly approximated as a prism:



After developing a lower bound on our differential equation, the model proves surprisingly well! This is because our invariant holds through dl, and can be used to directly prove our lower bound.

### 3.2 Model 2: save rate model

The second step of the model was creating a system in which passengers could be saved. To achieve this, we modelled a fleet of rescue ships as a single ship moving back and forth from our sinking ship and the shore. As such, the following variables were used:

*sinking boat position* :  $bP$

*rescue fleet position* :  $rP$

*rescue fleet human capacity* :  $rQ$

*rescue fleet speed* :  $rS$

*rescue fleet velocity* :  $rV$

*number of passengers on sinking ship at start* : *passengers*

*number of passengers that have already been saved* : *survivors*

#### Assumptions and simplifications:

1. Rescue vehicles move with constant speed and can stop instantaneously. This assumption is reasonable because the distance between the shoreline and the sinking ship is most probably very large compared to the distance needed for acceleration. Furthermore, on open water, a rescue ship doesn't really have any efficient strategy other than to make a B-line full speed ahead for its target. So for simplicity, we can assume constant velocity.
2. All passengers on board are alive before the ship is underwater and they cannot be rescued once the ship gets submerged. This assumption is a little too strict in that in reality, people might be able to survive in submerged ships. But for the sake of simplicity here, we discard this possibility. This is acceptable because saving all people before the ship is underwater is optimal for a rescue action.
3. We assume that the loading and unloading process on the rescue fleet happens instantaneously. This is reasonable because the time it takes is generally small compared to the time it takes for the fleet to travel back and forth between the shore and the sinking ship.

It is important to note here that ship positions are with respect to the shore, which would be at distance 0. Since the rescue fleet is always between the sinking ship and the shore, a natural invariant arises :  $0 \leq rP \leq bP$  . Another invariant is that

$(rV = rS) \mid (rV = -rS)$ , as we always model the rescue vehicles at their peak velocity since the open water allows them to achieve maximum speeds.

The key requirement for this model was that we need enough time at the start to rescue everyone on the ship. This means the rescue fleet must be able to make enough trips to get everyone back to shore, given rescue vehicle speeds, rescue vehicle carrying capacity, and boat position with respect to the shore!

Since it takes  $bP / rS$  seconds to go between the sinking ship and we must make  $passengers / rQ$  trips to save everyone, we thus have the bound that

$$T \geq bP/rS + (passengers/rQ) * 2 * bP/rS$$

For an invariant of this system therefore, we must prove the following always holds

$$T - t \geq (-rP + bP * rV/rS) / rV + ((passengers - survivors)/rQ) * 2 * bP/rS$$

As long as this invariant holds, we know that we have enough time to save all passengers regardless of where the rescue fleet currently is!

Additional assumptions about the combined model:

The rescue operation doesn't affect the sinking process. Normally, mass of the ship is much larger than the total mass of passengers, so even though the rescue operation reduce the number of passengers on board, it doesn't make much difference to the mass of the ship and thus wouldn't considerably affect the sinking process.

## 4 KeYmaeraX — How we went about proving

### 4.1 The Sink Rate Model

To prove a lower bound on the sinking time, we found a differential invariant by solving the differential equation, and then apply the differential invariant with necessary differential cuts to prove the model.

The dynamics of this model is given as:

$$\text{waterOnBoard}' = \text{hole} \quad (1)$$

$$d' = dV \quad (2)$$

$$dV' = (m + \text{waterOnBoard} * p - p * A * d) * g/m \quad (3)$$

Intuition:

Looking at (2) and (3), we figured that the relation between  $d$  and  $dV$  is very similar to the  $x, y$  coordinate in circular motion, where  $x' = y$  and  $y' = -x$ . Thus, it might be possible to find a differential invariant that looks like the circular motion invariant  $x^2 + y^2 = R^2$ .

Step1: Solve differential equations

Differentiate equation (3):

$$dV'' = (\text{waterOnBoard}' * p - p * A * d') \quad (4)$$

Substitute in (1) and (3):

$$dV'' = (\text{hole} * p - p * A * dV) \quad (5)$$

Solving (5):

$$dV(t) = B * \sin\left(\sqrt{\frac{p * A * g}{m}} * t + \phi\right) + \frac{\text{hole}}{A} \quad (6)$$

Where  $B$  and  $\phi$  are constants.

Plug (6) back into equation (3) to solve for  $d$ :

$$d(t) = -\sqrt{\frac{m}{p * A * g}} * B * \cos\left(\sqrt{\frac{p * A * g}{m}} * t + \phi\right) + \frac{\text{hole}}{A} * t + \frac{m}{p * A} \quad (7)$$

Step2: Spot differential invariants

Equation (6) contains the sine function and equation (7) contains the cosine function. Thus, by applying trigonometric identity  $\sin^2(\theta) + \cos^2(\theta) = 1$ , we obtained the differential invariant:

$$\left(\sqrt{\frac{p \cdot A \cdot g}{m}} * \left(d - \frac{\text{waterOnBoard}}{A} - \frac{m}{p \cdot A}\right)\right)^2 + \left(dV - \frac{\text{hole}}{A}\right)^2 = B^2 \quad (8)$$

Since (8) contains the variable *waterOnBoard*, we also need a differential invariant for it, which is  $\text{waterOnBoard} = \text{hole} * t$ .

Step3: Derive a lower bound on sinking time

Before the ship completely sinks down, depth under water *d* is smaller than or equal to the height *h*. So  $d \leq h$ . From equation (7), we derived the inequality:

$$d(t) \geq \sqrt{\frac{m}{p \cdot A \cdot g}} * B + \frac{\text{hole}}{A} * t + \frac{m}{p \cdot A}$$

Note: This inequality can also be derived from our differential invariant, equation (8), which means we have chosen the correct invariants.

Combining these two inequalities, the lower bound on *t* is given as:

$$t \geq \left(h - B * \sqrt{\frac{m}{p \cdot A \cdot g}} - \frac{m}{p \cdot A}\right) * A / \text{hole}$$

Step4: Prove in KeYmaera X

Originally, we proved this model on its own (without any loop) by applying monotonicity, differential cut and differential invariant. But then in order to integrate the sink rate model into the save rate model, we needed to loop the dynamics. To prove that, we introduced a loop invariant, which is just the differential invariant, and then everything else was proved in the same way.

## 4.2 The Save Rate Model

To prove the amount of time sufficient enough to save all passengers on board, we had the following loop invariant:

$$\begin{aligned} &(\text{survivors} = \text{passengers}) \\ &\text{OR} \\ &(\text{survivors} < \text{passengers} \ \& \ t < T \ \& \ (rV=rS \ | \ rV=-rS) \ \& \ T-t \geq (-rP+bP*rV/rS)/rV + \\ & \quad ((\text{passengers}-\text{survivors})/rQ)*2*bP/rS) \end{aligned}$$

Essentially, either everyone has already been saved, or they haven't. If they haven't, then we haven't exceeded our time limit, our system is valid, and the time remaining is sufficient to complete the rescue mission. " $t < T$ " is actually proven from " $T - t \geq (-rP + bP * rV / rS) / rV + ((\text{passengers} - \text{survivors}) / rQ) * 2 * bP / rS$ ". To prove this latter invariant, it is sufficient to either ghost in a variable for "single loop time" to solve for  $rP$ . It is also possible to DI the equation; however, the interesting aspect of this DI prove is actually proving it holds in the first place! Since only ' $rP$ ' and ' $t$ ' are ever changing during our dynamics, the dl of this equation becomes  $-1 \geq -1$  based on whatever  $rV$  is. Proving this equation holds originally is strange because we don't know whether we are moving in the same direction! When  $rV_{old} = rV_{new}$ , things are simple because the equation is identical. When  $rV_{old} = -rS$  and  $rV_{new} = rS$ , things are also simple because we're basically saying 'given that we were moving away from the target and we had enough time to return, now that we are returning we definitely have enough time.' The stranger aspect to prove is when we have  $rV_{old} = rS$  and  $rV_{new} = -rS$ , where we want to prove that even though we are leaving the sinking ship, we will have enough time to return! What makes this step solvable is that this only can happen if we are currently at the sinking ship. Thus,  $rP = bP$  when we are changing velocity. Thus, our model from the previous loop states

$$(1) \quad T - t \geq (-rP + bP * (rS) / rS) / (rS) + ((\text{passengers} - \text{survivors}) / rQ) * 2 * bP / rS$$

While we must prove that after picking up new survivors and heading back to shore

$$(2) \quad T - t \geq (-rP + bP * (-rS) / rS) / (-rS) + ((\text{passengers} - (\text{survivors} + rQ)) / rQ) * 2 * bP / rS$$

If we subtract equation 1 from equation 2 and simplify, we find

$$0 \geq 2 * rP / rS - 2 * bP / rS$$

This equation holds only if  $bP \geq rP$ , but since we are at the ship our model only tells us that  $rP \geq bP$ ! Since our domain constraint is  $0 \leq rP \leq bP$ , however, we know that  $rP > bP$  does not satisfy this and thus is true by blow up.

### 4.3 The Combined Model

Our third and final model proves that we can link up our two previous models and have them work together! The length of such a proof required diligence and patience, as KeYmaeraX and mathematica become slower as the proof continues. Originally, we were hoping to fully integrate the models to say " $t < (h - B / w - m / (p * A)) * A / \text{hole}$ " instead of " $t < T$  &  $T = (h - B / w - m / (p * A)) * A / \text{hole}$ "; however, KeYmaeraX / mathematica would consistently freeze on this model. To make the solving process simpler, we kept " $T = (h - B / w - m / (p * A)) * A / \text{hole}$ ". First, we prove the Sink Rate submodel; after this proof, we can then abstract away what " $T$ " really is an essentially hide any equation relating to

g,p,A,m,etc. The act of abstracting these equations eases computation for KeYmaeraX and allows us to more smoothly proof our second submodel — the Save Rate submodel.

## 5 Conclusion (goals met / not met)

In total, we met our goals of proving a sensible sink rate lower bound, showing that we can save everyone in this given time period, and having a single unified model of the entire system. The only problem we didn't fulfill was proving this model on more complex ships; **even with an upright prism, the model was not completely solvable and could only be bounded.** We met with Brandon in Office Hours on multiple occasions, but still found it too difficult to prove models using hemispheres and sideways non-rectangular prisms.

In the end, what our model shows is this constraint is all that is needed to save everyone in time:

$$T = (h - B/w - m/(p * A)) * A/hole \geq bP/rS + (passengers/rQ) * 2 * bP/rS$$

This allows us to clearly see the relationship between various variables of our system! We need less time to save everyone if our ship is tall or wide (everything else constant), however we need more time if our ship is heavy or the hole in it is big. Also shown, the more dense our suspension liquid is, the stronger buoyancy becomes and thus the more time we have to save! On the other hand, faster speeds, save rate, and closeness to the shore allow people to be saved faster.

In comparing our model with the typical state of the art model, we are using logic instead of simulations! While others may simulate a ship sinking, we prove a bounded time on how long it will take the ship. Others focus specifically on the ship falling apart or tilting, but we focus on the pure force model.

## 6 Project Deliverables

Included in our project deliverables is a proof of our Sink Rate model (without loop: "forever\_sinking.kya", with loop: "forever\_sinking\_loop.kya"), proof of our Save Rate model ("save\_passengers.kya"), proof of our Combined model ("full\_model.kya"), and a Python2 Tkinter simulation of the model (which also outputs \*.csv files of relevant variables to be opened in excel).

Irene Li solved the differential equation needed to prove the Sink Rate model; Jordan Tick proved the Save Rate model and created the Tkinter simulation of the system. We worked together to solve the final model.