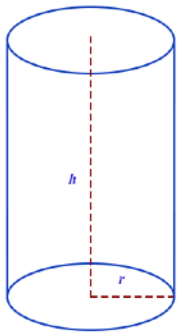
PROBLEM 1: BEER CAN SIZE

180 billion aluminium beverage cans are produced annually. Optimising the shape of the can to minimise the material use yields huge benefits in terms of raw material used. Your goal is to find the optimal can dimensions assuming the can is a perfect cylinder.



Plot the volume and surface of the can as a function of the cylinder radius. Try to find the optimum can dimensions by looking at the plots. Find the optimal radius and height for a 0.33 and 0.5l cans.

Relevant functions:

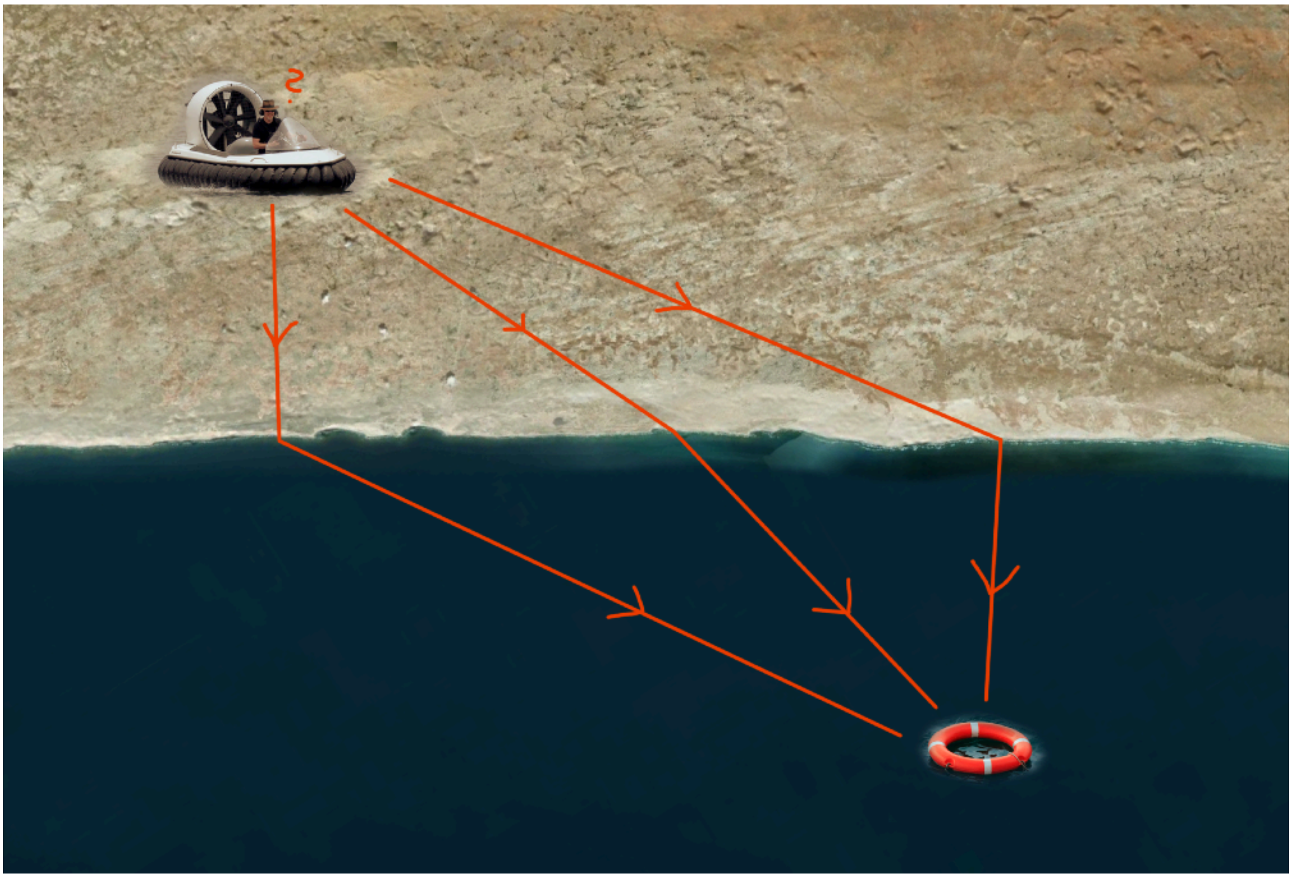
```
plot
line
subplot
linkaxes
```

PROBLEM 2: AUTOMATE THE OPTIMUM SEARCH

Finding optimum point by calculating all possible configurations is typically not possible because of the size of the domain and complexity of evaluating the cost function. For example, simulating the performance of a one design of a diesel engine can take hours on supercomputer clusters. You will therefore need to find a better way of finding the optimum. Use the built-in optimisation functions to solve the can problem.

Relevant functions:

```
fmincon
```



PROBLEM 3: HOVERCRAFT TRAVEL TIME

Hovercrafts can travel on both water and land. Due to reasons the speeds on land and water differ. On land it can travel 7m/s but on water the speed it is limited to 4m/s. You are chilling in your hovercraft on a desert and you suddenly see someone drowning in the water. You need to act fast! You crack open your laptop to find the fastest way to reach the drowning person.

Given some initial coordinates on land and water what is the travel time from one point to the other? Assume you drive in a straight line from your initial position on land to a location on the beach and then you drive in a straight line on water. Your travel time will depend on the point where you enter the water.

Write a MATLAB function that will calculate the travel time. Show it to your favourite MATLAB teacher.

Plot the travel path on a *xy* plot.

Use the `fmincon` to solve the problem and finally save that person.

Once you have the function written check if it can solve the problem for multiple water entry points at the same time. If not try to update it.

PROBLEM 4: MUD IN THE WAY

You travel with your favourite hovercraft to Florida. In Florida there are muddy shores that are hard to cross with a hovercraft: 2m/s. Now, to save someone you will need to cross an additional band of mud. Update your hovercraft travel function to allow more complex terrain layer geometries.

PROBLEM 5: OPTIMISING THE MUDDY RESCUE PATH

Optimising the travel path through muddy banks requires us to find two points: where to enter the mud and where to enter the water. To solve this problem implement a simple differential evolution algorithm.

> “DE optimises a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple formulae, and then keeping whichever candidate solution has the best score or fitness on the optimisation problem at hand. In this way the optimisation problem is treated as a black box that merely provides a measure of quality given a candidate solution and the gradient is therefore not needed.”

Pseudocode:

```
// xi vector of the current population
// yi vector of the new population

while (convergence criterion not yet met) {
  for (i=0; i<NP; i++)
  {
    r1 = rand(NP);
    r2 = rand(NP);
    r3 = rand(NP);
    ui = x(r3) + F*(x(r1) - x(r2));
    if (f(ui) <= f(xi)) {
      yi = ui; }
    else {
      yi = xi; }
  }
}
```