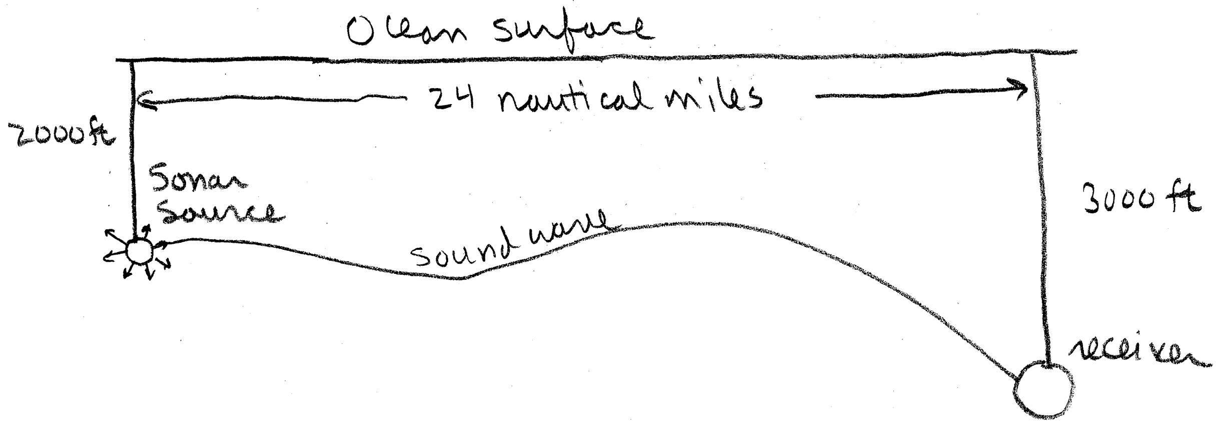
Application of Root Finding Methods



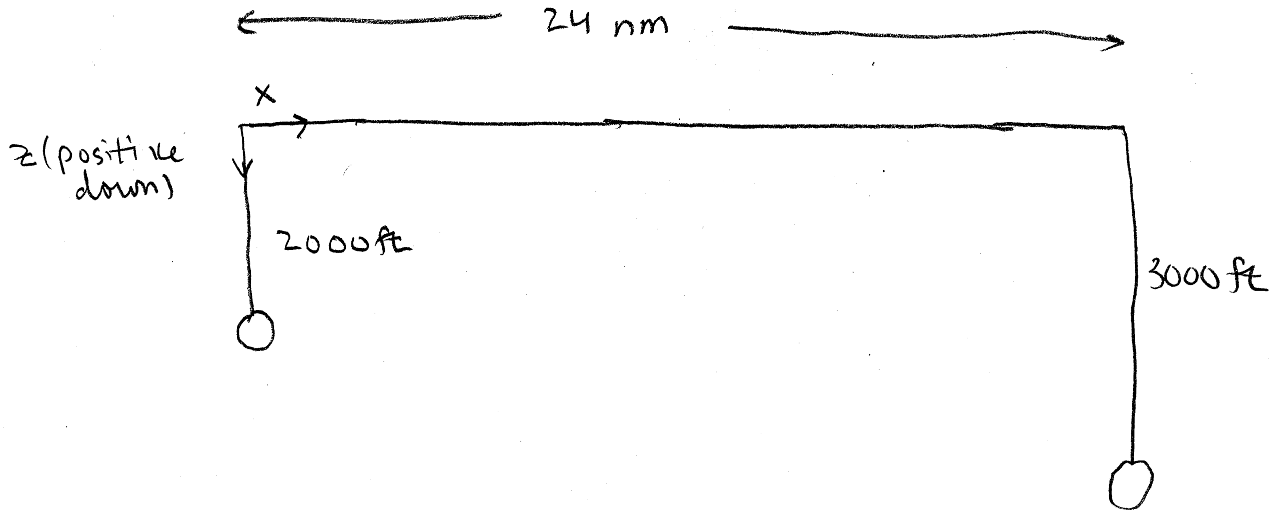
- The sonar source emits sound waves. Which of these waves reach the receiver?

- If the density of the ocean was constant, this problem would be trivial. However, the density varies and this complicates the problem

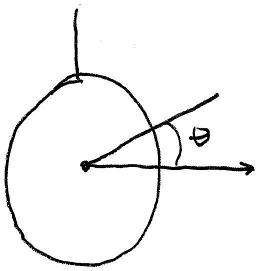
- Main Assumptions

  * although the source emits sounds in a 3-D spherical pattern, we will assume a 2D geometry

  * The speed of sound in the ocean is a function only of depth. This is known only as a table of values.



Sonar Source



θ = angle of departing signal

For which angles (θ) does the sonar signal reach the receiver?

Input: angle θ

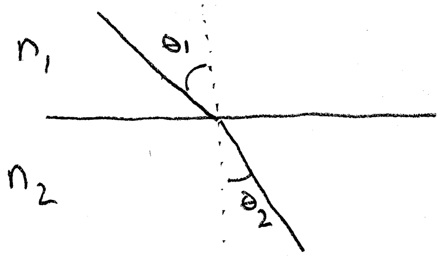
Output: Depth of signal at $x = 24$ nautical miles ($D(\theta)$)

This is our function. Want to find θ s.t.

$D(\theta) = 3000$ or $\underbrace{D(\theta) - 3000 = 0}_{\text{function to find the roots of.}}$

Problem: $D(\theta)$ is messy.

- The key physical principle is Snell's Law



A ray (light, sound, any wave) gets deflected any time it passes through a change in the index of refraction (n).

$$\frac{\sin(\theta_1)}{n_1} = \frac{\sin(\theta_2)}{n_2}$$

The value of n changes with density

- we need a continuous version of this equation

⇝ after much work

z = depth of ray as a function of horizontal distance from source (x)

c = speed of sound in ocean as a function of z.
$= c(z)$

$$c^4 \frac{d^2 z}{dx^2} + \left(3\frac{dc}{dz} - c\frac{d^2 c}{dz^2}\right)\frac{dz}{dx} = 0$$

c is known, but only as a table of values.

$$z(x=0) = 2000$$

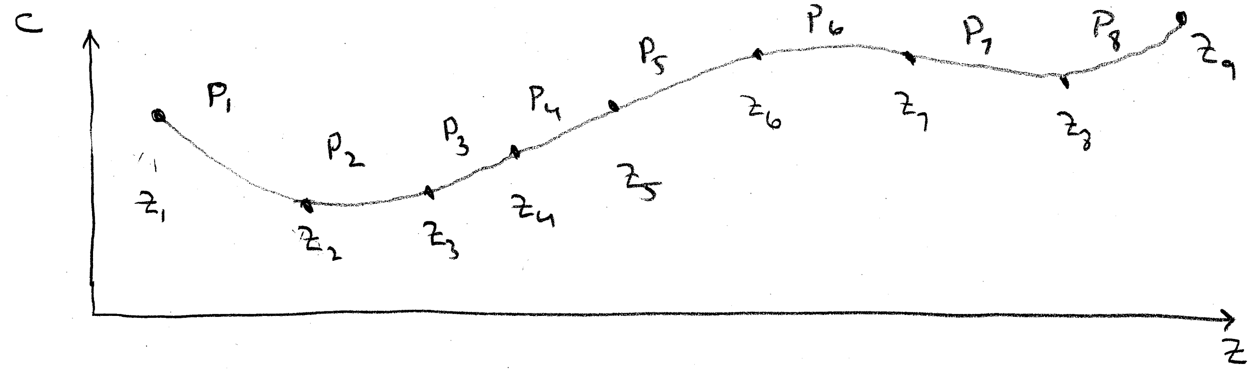
$$\frac{dz}{dx}(x=0) = \tan(\theta)$$

- To evaluate $D(\theta)$, solve the above 2^{nd} order differential equation for $z(x)$, then compute $z(24)$.

- C is known only from a table of values, but in order to solve the differential equation, we need to be able to get the value of C at any z. (also C', C'')

- The solution to this is <u>interpolation</u> (ie, using known values to predict unknown values).

- The general technique in a case like this is to use <u>cubic splines</u>



* Between 2 points, a cubic polynomial is created (p_1 through p_8 in the figure above)

* These are constructed so that
    - The values of the p's match at endpoints
    - The values of p' match at the endpoints
    - The values of p'' match at the endpoints
    - For example

$p_1(z_2) = p_2(z_2)$ continuity of p

$p_1'(z_2) = p_2'(z_2)$ continuity of p'

$p_1''(z_2) = p_2''(z_2)$ continuity of p''

this ensures that you get a smooth curve between data points.

* There are subroutines that will do this for you.