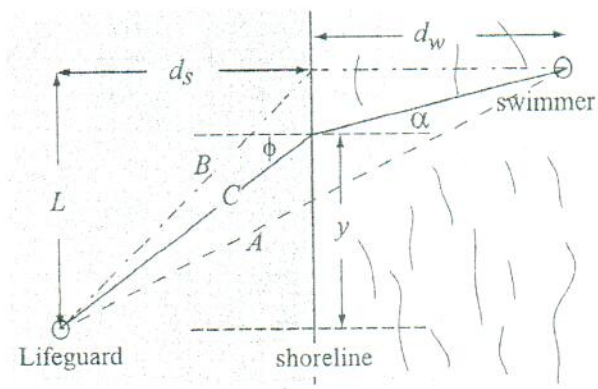
Lifeguard Duty -- Finding the Optimum Path

A student has a summer job as a lifeguard at the beach. After spotting a swimmer in trouble, the lifeguard tries to deduce the path by which he/she can reach the swimmer in the shortest time. The path of shortest distance (path A) is obviously not best since it maximizes the time spent swimming (the lifeguard can run faster in the sand than he/she can swim in the water). Similarly, path B, which minimizes the time spent swimming, is probably not the best choice, since it is the longest reasonable path. Thus, the optimal path is probably somewhere between paths A and B.



Consider an arbitrary intermediate path (such as path C in the diagram) and the following data for a particular situation:

speed in sand: $v_s = 3$ m/s speed in water: $v_w = 1$ m/s

appropriate distances (see diagram): $L = 48$ m, $d_s = 30$ m, $d_w = 42$ m

where y is the lateral distance along the shoreline at which the lifeguard enters the water.

Now, with the above data and description of the problem, develop expressions for the total distance traveled by the lifeguard and the travel time as a function of the lateral distance y. Plot these functions on different axes on the same page using Matlab's ***subplot*** command. Do the distance vs. y and travel-time vs. y relationships make sense? What value of y gives the minimum distance traveled (i.e. path A)? What is the optimum y if the travel time is to be minimized? Clearly develop the logic for your analysis and rationalize that the resulting plots are reasonable solutions to this simple geometry problem…

from similar Triangles $\frac{L}{d_s + d_w} = \frac{L - y_{min}}{d_w}$ or $y_{min} = L - \frac{d_w L}{d_s + d_w}$ ← value of y for path A

$y_{max} = L$ ← value of y for path B

Now for any y between the min and max values, we have

distance traveled in water $l_w = \sqrt{(L-y)^2 + d_w^2}$

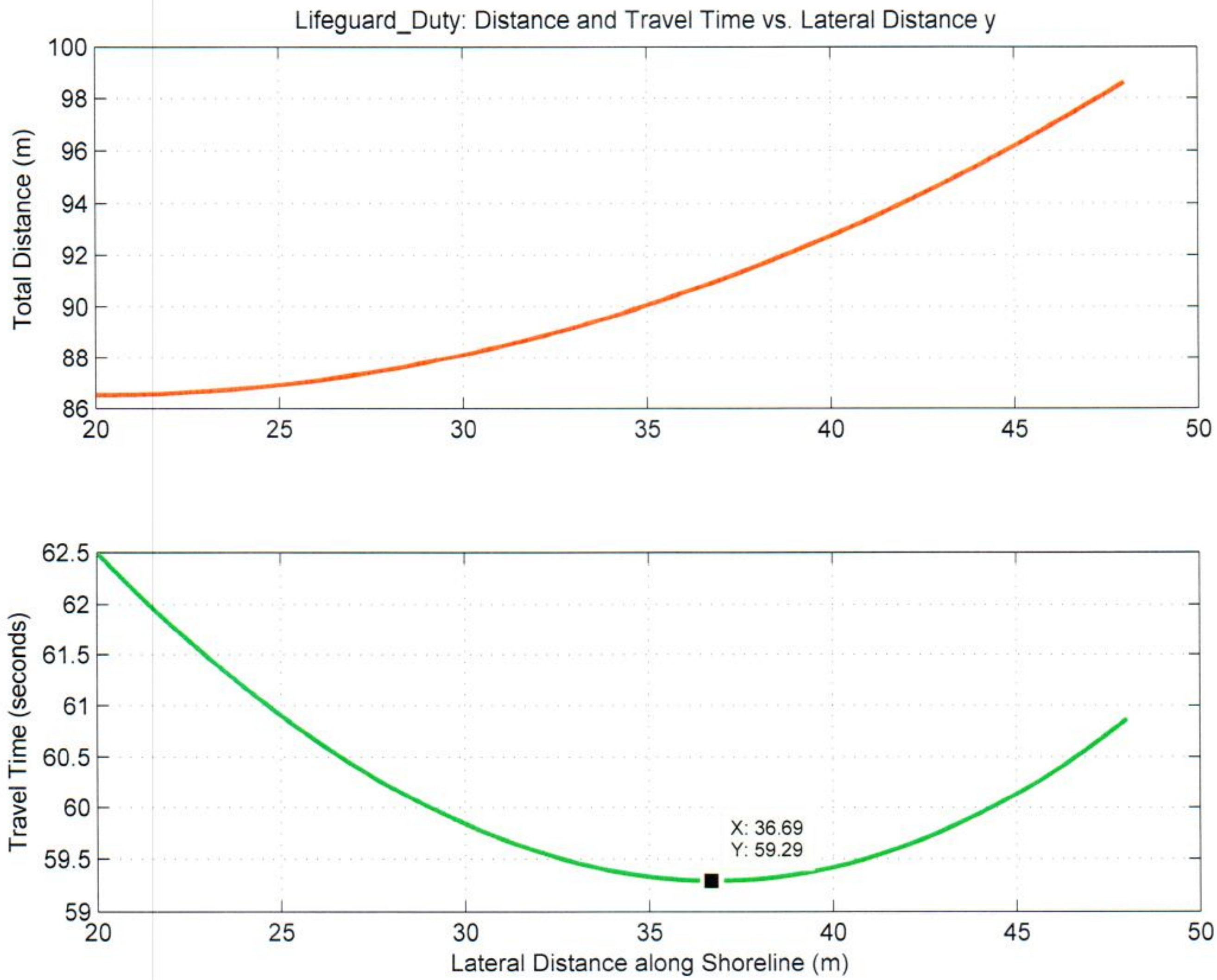
distance traveled in sand $l_s = \sqrt{y^2 + d_s^2}$

∴ total distance Traveled $l_{TOT} = l_w + l_s$

and Total Travel time $T_{TOT} = \frac{l_w}{v_w} + \frac{l_s}{v_s}$

} plot these two functions vs. y

see lifeguard_duty.m


Lifeguard_Duty: Distance and Travel Time vs. Lateral Distance y
Total Distance (m)
100
98
96
94
92
90
88
86
20
25
30
35
40
45
50
Travel Time (seconds)
62.5
62
61.5
61
60.5
60
59.5
59
X: 36.69
Y: 59.29
Lateral Distance along Shoreline (m)

```
%
%   LIFEGUARD_DUTY.M     Plots the distance and travel time for a lifeguard vs.
%                      lateral distance along shoreline where lifeguard enters water
%
%   This program again illustrates how to evaluate and plot functions in Matlab. In
%   this case, we evaluate the distance traveled and the travel time for a lifeguard
%   as a function of the lateral distance along the shoreline (see problem
%   description).  Going in a direct straight line from the lifeguard to the swimmer
%   is the shortest path, but it takes longer to swim a given distance than to run
%   the same distance in the sand.  Thus, there may be a trade off if the shortest
%   travel time is desired -- taking a somewhat longer total path to shorten the
%   total travel time.  Let's see if this is true...
%
%   Reference:  This problem is based on Prob. 3 in Chapt 4 of the 3rd Ed of the book,
%   Matlab - An Introduction With Applications, by A. Gilat (2008).
%
%   File prepared by J. R. White, UMass-Lowell (Sept. 2017)
%

      clear all,  close all,   nfig = 0;
%
%   let's set some parameters for the problem
      dw = 42;   ds = 30;       % horizontal distances (m) in water (dw) and sand (ds)
      L = 48;                   % max vertical distance (m) along shoreline
      vw = 1;  vs = 3;          % speeds (m/s) in water (vw) and sand (vs)
      ymin = L - dw*L/(ds+dw); % minimum lateral distance (m)
%
%   compute distances traveled  Lw -> water  and  Ls -> sand
      y = linspace(ymin,L,100);     % vector of lateral distances
      Lw = sqrt((L-y).^2 + dw^2);  % vector of distances in water
      Ls = sqrt(y.*y + ds^2);       % vector of distances in sand
      Ltot = Lw + Ls;               % total distance traveled (m)
%
%   now compute the time traveled for each y value
      Ttot = Lw/vw + Ls/vs;         % travel time (seconds)
      [Tmin,i] = min(Ttot);  y_Tmin = y(i);  % min time and corresponding y value
      disp('minimum time (seconds) & corresponding y value (m):'); disp([Tmin y_Tmin])
%
%   plot results
      nfig = nfig+1;   figure(nfig)
      subplot(2,1,1),plot(y,Ltot,'r-','LineWidth',2), grid
      title('Lifeguard\_Duty: Distance and Travel Time vs. Lateral Distance y');
      ylabel('Total Distance (m)')
      subplot(2,1,2),plot(y,Ttot,'g-','LineWidth',2), grid
      xlabel('Lateral Distance along Shoreline (m)'),ylabel('Travel Time (seconds)')
%
%   end of program
```

Simple Cooling tank

Solve $\frac{dT}{dt} = -\frac{UA}{mc}(T - T_\infty)$ with $T(0) = T_h$

for the time the system takes to decrease the water temperature to T_c.

— —

for ease of manipulation, let $k = \frac{UA}{mc}$

then $\frac{dT}{T - T_\infty} = -k\,dt$ ← separable

let $u = T - T_\infty$ and $du = dT$

$$\therefore \int_{T_h}^{T_c} \frac{du}{u} = -k \int_0^{t_f} dt = -k t_f$$

$$\ln u \Big|_{T_h - T_\infty}^{T_c - T_\infty} = \ln(T_c - T_\infty) - \ln(T_h - T_\infty)$$

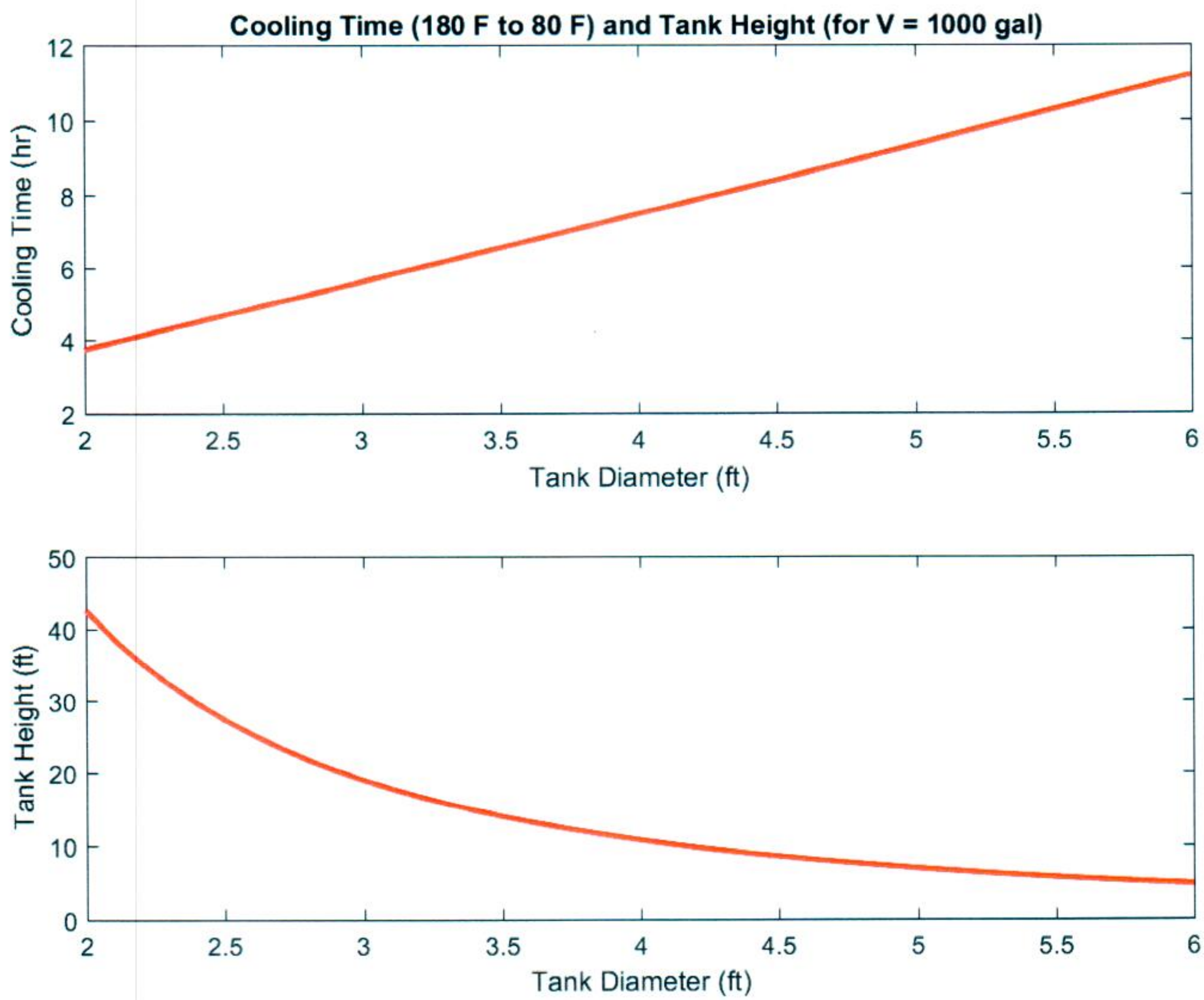
$$= \ln \frac{T_c - T_\infty}{T_h - T_\infty}$$

$$\therefore t_f = -\frac{1}{k} \ln \frac{T_c - T_\infty}{T_h - T_\infty}$$

or

$$\boxed{t_f = -\frac{mc}{UA} \ln \frac{T_c - T_\infty}{T_h - T_\infty}}$$ ans

↑ this is the time it takes to cool from T_h to T_c


Cooling Time (180 F to 80 F) and Tank Height (for V = 1000 gal)
Cooling Time (hr)
12
10
8
6
4
2
2
2.5
3
3.5
4
4.5
5
5.5
6
Tank Diameter (ft)
Tank Height (ft)
50
40
30
20
10
0
2
2.5
3
3.5
4
4.5
5
5.5
6
Tank Diameter (ft)

```
%
%   COOLING_TANK_1.M   Perform analysis of a simple Cooling Tank
%
%   This program illustrates how to evaluate and plot functions in Matlab. In
%   this case, we evaluate the cooling time for a holding tank versus the diameter
%   of the tank (for constant volume).  We also look at the tank height vs diameter
%   so that a reasonable tank size can be selected.
%
%   Reference:  This problem is based on Example 10.3 in the book Numerical Methods
%   with Chemical Eng. Applications by Al-Malah (2014)
%
%   File prepared by J. R. White, UMass-Lowell (last update: Sept. 2017)
%

      clear all,  close all,   nfig = 0;
%%
%   set problem parameters
      Vg = 1000;          % tank/liquid volume (gal)
      cf = 7.48052;       % conversion factor (7.48052 gal per ft^3)
      Vcf = Vg/cf;        % tank/liquid volume (ft^3)
      Th = 180;           % hot temperature (F)
      Tc = 80;            % cold temperature (F)
      Tinf = 70;          % environmental temp (F)
      c = 1.00;           % specific heat of water (BTU/lbm-F)
      rho = 62.2;         % water density (lbm/ft^3)
      m = rho*Vcf;        % water mass (lbm)
      U = 20;             % overall HT coeff (BTU/hr-ft^2-F
      D = 2:0.1:6;        % vector of tank diameters (ft)
      H = 4*Vcf./(pi*D.^2);  % vector of tank heights (ft) (for constant volume)
      A = pi*D.*H;        % vector of HT areas (ft^2)
%
%   compute the cooling times and tank heights
      tf = -(m*c/U./A)*log((Tc-Tinf)/(Th-Tinf));
%%
%   plot results --  tf vs D
      nfig = nfig+1;   figure(nfig)
      subplot(2,1,1), plot(D,tf,'r-','LineWidth',2), grid
      title(['Cooling Time (',num2str(Th),' F to ',num2str(Tc), ...
             ' F) and Tank Height (for V = ',num2str(Vg),' gal)']);
      xlabel('Tank Diameter (ft)'),ylabel('Cooling Time (hr)')
      subplot(2,1,2), plot(D,H,'r-','LineWidth',2), grid
      xlabel('Tank Diameter (ft)'),ylabel('Tank Height (ft)')
%
%   Note: Based on the plots, a smaller diameter gives a shorter cooling time.
%         However, the tank height is unreasonably large for small diameters (for V =
%         1000 gal).  Thus, a tank diameter in the 3.5 - 4.5 ft range is probably the
%         best compromise. D = 3 ft would give a cooling time of under 6 hrs, but the
%         H would be close to 20 ft -- this  is a pretty tall and skinny tank design!!!
%
%   end of program
```