

**1.1**

- (a) Fetch is the distance over the surface of the water on which a wind blows.
- (b) In a dispersive medium the speed at which a wave travels depends on its wavelength.

1.2 According to the table on page 9 the most common wave height is 3-4 feet.

1.3 Starting with  $L = \frac{g}{2\pi}T^2$  this becomes

$$\frac{2\pi}{k} = \frac{g}{2\pi} \left( \frac{2\pi}{\omega} \right)^2$$

$$\frac{1}{k} = \frac{g}{\omega^2}$$

$$\omega^2 = gk$$

While  $c = \sqrt{\frac{gL}{2\pi}}$  becomes

$$c = \sqrt{\frac{g}{k}}$$

$$c^2 = \frac{g}{k}$$

Now the first of these gives  $\omega = \sqrt{gk}$  so that

$$\omega^2 = gk$$

$$\rightarrow \omega = \sqrt{gk}$$

$$\rightarrow \frac{\omega}{k} = \frac{\sqrt{gk}}{k} = \sqrt{\frac{g}{k}} = c$$

**2.1** Use the definition of pressure:

$$P_{\text{elephant}} = \frac{F}{A} = \frac{mg}{2\pi r^2} = 1.6 \times 10^5 \text{ Pa}$$

$$P_{\text{woman}} = \frac{F}{A} = \frac{mg}{\pi r^2} = 117.0 \times 10^5 \text{ Pa}$$

**2.2** The elephant will apply a force equal to its weight  $mg$ . Thus in order for the air inside to support the elephant it must be at a pressure of at least  $P = F/A = mg/A = (1000\text{kg})g/(1.5\text{m} \times 2.0\text{m}) = 3,267 \text{ Pa}$ . We need to compare this with the pressure we are able to produce (0.5atm). Converting we find that  $0.5\text{atm} = 50,600\text{Pa}$ . Thus one elephant will not be a problem.

$$2.3 \quad \Delta P = \rho g h \quad \rightarrow \quad h = \frac{\Delta P}{\rho g} = \frac{0.5(1.013 \times 10^5 \text{ Pa})}{(1000 \text{ kg/m}^3)g} = 5.2 \text{ m}$$

**2.4** Since the box of water is not moving we know that the acceleration is zero. Thus we know that  $\Sigma \vec{F} = 0 \rightarrow \Sigma F_y = 0$ . Thus let us add the vertical components of the forces acting on the box of water. The forces on the sides of the box because of the pressure of the water have no vertical component so we just have the force due to the pressure on the top and bottom ( $F_{\text{top}}$  and  $F_{\text{bottom}}$ ) and the force of gravity ( $F_G$ ). Thus

$$\begin{aligned} \Sigma F_y &= 0 \\ F_{\text{bottom}} - F_{\text{top}} - F_G &= 0 \\ F_{\text{bottom}} - F_{\text{top}} - [mg] &= 0 \\ [P_{\text{bottom}}A_{\text{bottom}}] - [P_{\text{top}}A_{\text{top}}] - [\rho V]g &= 0 \\ P_{\text{bottom}}[bc] - P_{\text{top}}[bc] - \rho[hbc]g &= 0 \\ P_{\text{bottom}} - P_{\text{top}} - \rho hg &= 0 \\ P_{\text{bottom}} - P_{\text{top}} &= \rho gh \\ \Delta P = P_{\text{top}} - P_{\text{bottom}} &= -\rho gh = -\rho g \Delta y \end{aligned}$$

**2.5** First convert height to meters:  $730\text{ft} \times \frac{1\text{m}}{3.28\text{ft}} = 223\text{m}$ . Now we note that the pressure difference between the water side and the air side is about the same as the difference between the surface of the water and at a depth of 223 meters below the surface.

$$\Delta P = -\rho g \Delta y = -(1000\text{kg/m}^3)g(-223\text{m}) = 2.23 \times 10^5\text{Pa}$$

$$F = \Delta P A = (2.23 \times 10^5\text{Pa})(1.0\text{m}^2) = 2.23 \times 10^5\text{N}$$



**2.6** Use the pressure equation:

$$\begin{aligned}\Delta P &= -\rho g \Delta y \\ P_{\text{mtn}} - P_{\text{sea}} &= -\rho g [y_{\text{mtn}} - y_{\text{sea}}] \\ &= -(1.29 \text{kg/m}^3)g(1173 \text{m}) = -1.48 \times 10^4 \text{Pa}\end{aligned}$$

For going underwater we find the following.

$$\begin{aligned}\Delta P &= -\rho g \Delta y \\ P_{\text{sea}} - P_{\text{under}} &= -\rho g [y_{\text{sea}} - y_{\text{under}}] \\ &= -(1025 \text{kg/m}^3)g(8854 \text{m}) = -1.18 \times 10^7 \text{Pa}\end{aligned}$$

**2.7** The pressure in the box will be the same as the pressure where the box was sealed. So assuming that the diver lives near the sea, the pressure inside is about the pressure of the air above the sea. The difference in pressure between the outside and inside of the box will be the same as the difference in pressure between the bottom of the sea and the surface of the sea.  $\Delta P = -\rho g \Delta y = -(1025 \text{kgm}^3)g(-200\text{m}) = 2.0 \times 10^6 \text{Pa}$  The force on one face of the box due to the difference in pressure is  $F = \Delta P A$ . There are three different size faces of the box, with areas 0.02, 0.03, and 0.06 square meters. So the forces are  $4.0 \times 10^4 \text{N}$ ,  $6.0 \times 10^4 \text{N}$ ,  $1.2 \times 10^5 \text{N}$ .

**2.8** Let  $P_A$  be the pressure in container  $A$ , and  $P_B$  the pressure in container  $B$ . The pressure at the top of the water on side  $A$  must be essentially the same as the pressure in container  $A$  because the density of air is so small. Thus we can say the following.

$$\begin{aligned}\Delta P &= -\rho g \Delta y \\ P_A - P_B &= -\rho g [y_A - y_B] \\ &= -\rho g [-10\text{cm}] \\ &= -(500\text{kg/m}^3)g[-0.10\text{m}] = 490\text{Pa}\end{aligned}$$

So we see that the pressure in container  $A$  is 490Pa greater than the pressure in container  $B$ .

**2.9** The volume rate of flow is the volume of water per second that passes through the stream. In a time  $dt$  the stream moves a distance of  $vdt$  thus a volume  $dV = Avdt$  (where  $A$  is the cross-sectional area of the stream) passes down the stream in a time  $dt$ . Thus the volume per time is simply  $\frac{dV}{dt} = Av$ . In this particular stream this gives a volume rate of flow of

$$\frac{dV}{dt} = Av = (4.2\text{m} \times 0.85\text{m})(0.62\frac{\text{m}}{\text{s}}) = 2.2\text{m}^3/\text{s}$$

Further down the stream has the volume rate of flow is the same (assuming that no water is added or lost from the stream). With  $d$  being the depth of the stream and  $w$  the width we can write the following.

$$\frac{dV}{dt} = A'v' = wd v' \longrightarrow d = \frac{dV/dt}{wv'} = 0.68\text{m}$$

**2.10** After the water leaves the nozzle it is under free fall so that we can use the projectile motion equations to find the initial velocity from the trajectory.

$$\begin{aligned}x_o &= 0 & y_o &= (1.0\text{m}) \\x_f &= (5.0\text{m}) & y_f &= 0 \\v_{o_x} &= v_o \cos 45^\circ = v_o/\sqrt{2} & v_{o_y} &= v_o \sin 45^\circ = v_o/\sqrt{2} \\a_x &= 0 & a_y &= -g\end{aligned}$$

Now consider the horizontal motion.

$$\begin{aligned}x_f &= x_o + v_{o_x}t + \frac{1}{2}a_x t^2 \\x_f &= 0 + \frac{v_o}{\sqrt{2}}t + 0 \\ \rightarrow \frac{t}{\sqrt{2}} &= \frac{x_f}{v_o}\end{aligned}$$

Now put this result into our equation for the vertical motion.

$$\begin{aligned}y_f &= y_o + v_{o_y}t + \frac{1}{2}a_y t^2 \\0 &= y_o + \frac{v_o}{\sqrt{2}}t - \frac{1}{2}gt^2 \\0 &= y_o + x_f - g\left(\frac{x_f}{v_o}\right)^2 \\v_o &= x_f \sqrt{\frac{g}{y_o + x_f}} = 6.4 \frac{\text{m}}{\text{s}}\end{aligned}$$

With this velocity we can find the volume rate of flow.

$$dV/dt = Av_o = \pi r^2 v_o = 8.0 \times 10^{-5} \text{m}^3/\text{s}$$

To spread 2.0 cm across an area of  $10\text{m} \times 10\text{m} = 100\text{m}^2$  we need a volume of water  $V = (0.02\text{m}) \times (100\text{m}^2) = 2.0\text{m}^3$ . Since  $V = \frac{dV}{dt} \Delta t$  we can calculate the time.

$$\Delta t = \frac{V}{dV/dt} = 2.5 \times 10^4 \text{s} = 6.9 \text{hours}$$

**2.11** The lift of the wing is due to the pressure difference between the top and the bottom. Since the pressure on the bottom is greater, the force on the bottom is greater.

$$F_{\text{lift}} = F_b - F_t = P_b A - P_t A = (P_b - P_t)A$$

With this relationship we can find the pressure difference.

$$P_b - P_t = F_{\text{lift}}/A = 1000\text{Pa}$$

But this pressure difference is caused by the different air speed on the top and bottom. Bernoulli's equation expresses this relationship.

$$\begin{aligned} \Delta[P + \frac{1}{2}\rho v^2 + \rho g y] &= 0 \\ (P_b - P_t) + \frac{1}{2}\rho(v_b^2 - v_t^2) + \rho g(y_b - y_t) &= 0 \\ (P_b - P_t) + \frac{1}{2}\rho(v_b^2 - v_t^2) + \rho g(0) &= 0 \\ \longrightarrow v_t &= \sqrt{\frac{2}{\rho}(P_b - P_t) + v_b^2} \\ &= 302.6 \frac{\text{m}}{\text{s}} \end{aligned}$$

Notice that this is less than one percent faster than the air on the bottom.

**2.12** We have two relationships for the flow in the tube; the equation of continuity,  $\Delta[vA] = 0$

$$v_w A_w - v_n A_n = 0 \quad \rightarrow \quad v_n = \frac{A_w}{A_n} v_w$$

and Bernoulli's equation.

$$\begin{aligned} \Delta\left[P + \frac{1}{2}\rho v^2 + \rho g y\right] &= 0 \\ (P_w - P_n) + \frac{1}{2}\rho(v_w^2 - v_n^2) + \rho g(y_w - y_n) &= 0 \\ (P_w - P_n) + \frac{1}{2}\rho(v_w^2 - v_n^2) + \rho g(0) &= 0 \\ (P_w - P_n) + \frac{1}{2}\rho(v_w^2 - v_n^2) &= 0 \end{aligned}$$

We have assumed that the pressure sensors are close together so that the difference in elevations between the two sensors is small;  $y_w - y_n \approx 0$ .

Combining the equation of continuity and Bernoulli's equation we find the following.

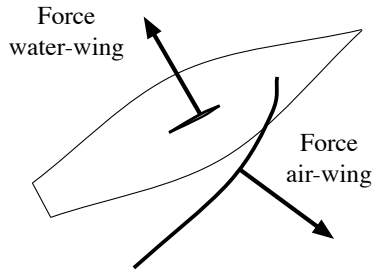
$$\begin{aligned} (P_w - P_n) + \frac{1}{2}\rho\left(v_w^2 - \frac{A_w^2}{A_n^2}v_w^2\right) &= 0 \\ (P_w - P_n) + \frac{1}{2}\rho\left(\frac{1}{A_w^2} - \frac{1}{A_n^2}\right)A_w^2 v_w^2 &= 0 \end{aligned}$$

From this we can solve for the volume rate of flow.

$$\frac{dV}{dt} = Av = A_w v_w = \sqrt{\frac{2(P_w - P_n)}{\rho\left(\frac{1}{A_n^2} - \frac{1}{A_w^2}\right)}}$$

With the given data this equation gives a volume rate of flow of  $1.45 \times 10^{-3} \text{m}^3/\text{s}$  or 1.45 liters per second.

2.13 We can see from the figure that the sum of the two forces is in the forward direction.





**2.14** Consider a point inside the house just under the roof, and a point outside the house just above the roof. These points are nearly at the same elevation so that  $\Delta y \approx 0$ .

$$\begin{aligned}\Delta\left[P + \frac{1}{2}\rho v^2 + \rho g y\right] &= 0 \\ (P_{\text{in}} - P_{\text{out}}) + \frac{1}{2}\rho(v_{\text{in}}^2 - v_{\text{out}}^2) + \rho g \Delta y &= 0 \\ (P_{\text{in}} - P_{\text{out}}) + \frac{1}{2}\rho(0^2 - v_{\text{out}}^2) + \rho g 0 &= 0 \\ \longrightarrow (P_{\text{in}} - P_{\text{out}}) &= \frac{1}{2}\rho v_{\text{out}}^2\end{aligned}$$

The net force due to the air pressure is

$$\begin{aligned}\vec{F}_{\text{net}} &= \vec{F}_{\text{in}} + \vec{F}_{\text{out}} \\ y : \quad F_{\text{net}} &= F_{\text{in}} - F_{\text{out}} \\ &= P_{\text{in}}A - P_{\text{out}}A \\ &= (P_{\text{in}} - P_{\text{out}})A \\ &= \frac{1}{2}\rho v_{\text{out}}^2 A = 6.0 \times 10^4 \text{N}\end{aligned}$$

This is positive so the net force is upward. The force is about ten times more than the weight of the roof, and thus it is harder hold the roof down than it is to hold the roof up.

**2.15** Let position 1 be in the small section and position 2 be in the large section. Let us start by using the continuity equation to find the velocity in the large section.

$$A_1 v_1 = A_2 v_2$$

$$\rightarrow v_2 = \frac{A_1}{A_2} v_1 = \frac{\pi r_1^2}{\pi r_2^2} v_1 = \left(\frac{r_1}{r_2}\right)^2 v_1 = \left(\frac{1}{2}\right)^2 v_1 = 2.0 \frac{\text{m}}{\text{s}}$$

Now we can use Bernoulli's Equation to find the pressure difference.

$$\Delta\left[P + \frac{1}{2}\rho v^2 + \rho g y\right] = 0$$

$$(P_2 - P_1) + \frac{1}{2}\rho (v_2^2 - v_1^2) + \rho g(y_2 - y_1) = 0$$

$$(P_2 - P_1) = -\frac{1}{2}\rho (v_2^2 - v_1^2) - \rho g(y_2 - y_1)$$

$$= -\left[\frac{1}{2}(v_2^2 - v_1^2) + g(y_2 - y_1)\right]\rho$$

$$= -\left[\frac{1}{2}\left((2\frac{\text{m}}{\text{s}})^2 - (8\frac{\text{m}}{\text{s}})^2\right) + g(3\text{m})\right](680\text{kg/m}^3)$$

$$= 480\text{Pa}$$

So we see that the pressure is 480 Pa greater in the large section than it is in the small section.

**2.16** We start with Bernoulli's equation.

$$\begin{aligned}
 \Delta \left[ \frac{1}{2} \rho v^2 + P + \rho g z \right] &= 0 \\
 \frac{1}{2} \rho (v_{\text{slow}}^2 - v_{\text{fast}}^2) + P_{\text{slow}} - P_{\text{fast}} + 0 &= 0 \\
 \longrightarrow P_{\text{slow}} - P_{\text{fast}} &= \frac{1}{2} \rho (v_{\text{fast}}^2 - v_{\text{slow}}^2) \\
 &= \frac{1}{2} \rho v^2 4\epsilon \\
 \longrightarrow \vec{F}_{\text{lift}} &= P_{\text{slow}} \vec{A}_{\text{slow}} + P_{\text{fast}} \vec{A}_{\text{fast}} \\
 &= P_{\text{slow}} A \hat{n} + P_{\text{fast}} A (-\hat{n}) \\
 &= (P_{\text{slow}} - P_{\text{fast}}) A \hat{n} \\
 \longrightarrow F_{\text{lift}} &= (P_{\text{slow}} - P_{\text{fast}}) A \\
 &= \frac{1}{2} \rho v^2 A 4\epsilon \\
 &= \frac{1}{2} \rho v^2 A C_L
 \end{aligned}$$

where  $\vec{A}$  is in the direction of the inward normal and  $\hat{n} = \frac{\vec{A}_{\text{slow}}}{A}$ .

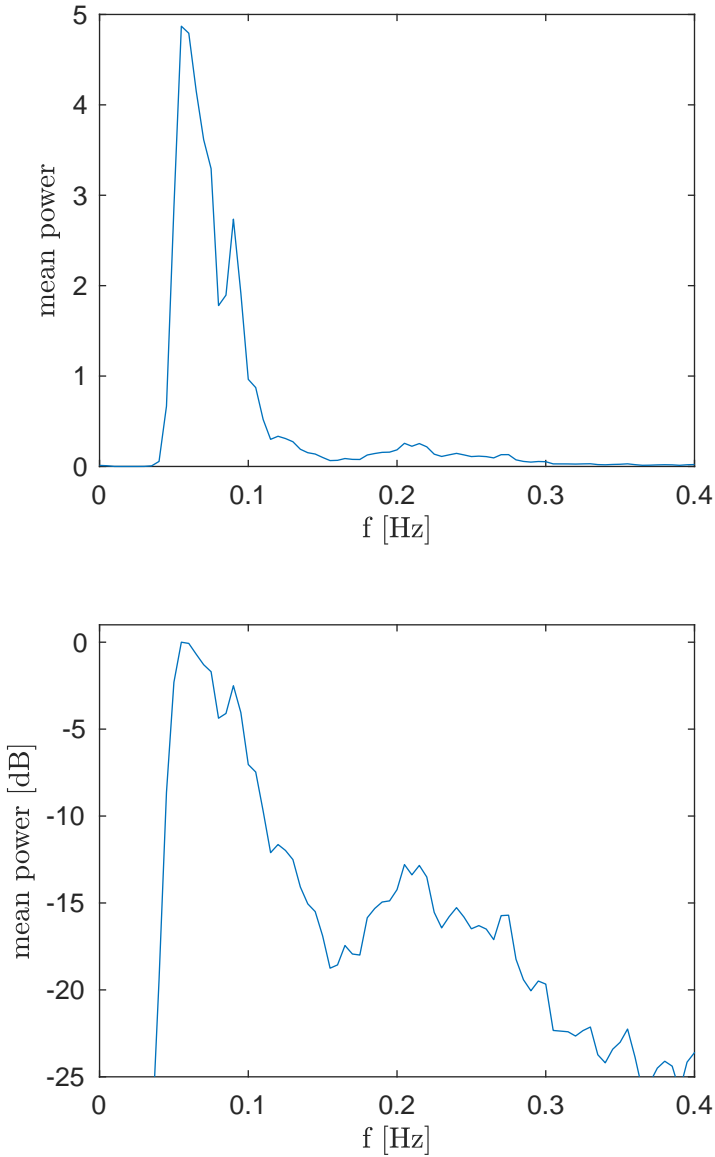
**3.1** Ship design is dependent on having a good estimate of the biggest wave one might encounter at sea. So engineers who design ships do require a more detailed description of the surface than is provided by the spectra analysis, and reason d is not justified.

**3.2** The most common techniques to measure waves at sea are based on either a buoy or a wave pole. The buoy measures its own vertical position as a function of time (typically every ? second) by measuring its own vertical acceleration (to be integrated twice to obtain the vertical position) or by using GPS. The vertical location of the sea surface along a wave pole is measured by measuring the change in electronic characteristics along a vertical wire at the sea surface (e.g., electrical resistance).

**3.3** With a buoy, directional wave information can be obtained by measuring the pitch-and-roll of the buoy (with inclinometers) or the yaw-and-sway of the buoy (with horizontal accelerometers or GPS). With a group of wave poles (at least three), directional wave information can be obtained by measuring surface slopes (in an array with small horizontal dimensions) or phase differences (in an array with larger dimensions).

**3.4** The instruments that are used to measure waves from satellites are the synthetic aperture radar (SAR) and the radar altimeter. The (synthetic) antenna of the SAR is large enough that it can distinguish individual waves. A SAR image therefore shows an image of individual waves (essentially in grey tone) but the transformation to a surface elevation is highly non-linear and has not yet been achieved. The antenna of an altimeter is much smaller than that of a SAR and the foot print at the sea surface is of the order of 1 kilometer in diameter, which is too large to distinguish individual waves (only an average roughness of the sea surface in the foot print can be estimated, from which the significant wave height can be inferred).

**3.5** The figure below is the result of breaking the data into chunks of 256 and doing the FFT for each of these chunks and then taking the average of the square of the amplitude at each frequency.



The following Matlab code will graph the power spectral density.

```
[1] function graphBuoyPSD()
[2]     S = load('buoy.mat');
[3]     z = S.buoy/100; % convert to meters since original data was in cm
[4]     fs = 1.28;
[5]     Nfft = 256;
[6]     [P,f] = myPSD(z,fs,Nfft);
[7]     figure(1)
[8]         plot(f,P)
[9]         xlim([0,0.4])
[10]        texText('f [Hz]', 'x')
[11]        texText('mean power', 'y')
[12]        graphToPDF(' ../buoyPSD.pdf',4,3)
[13]     figure(2)
```



```

[14]         dB = 10*log10(P/max(P));
[15]         plot(f,dB)
[16]         texText('f [Hz]', 'x')
[17]         texText('mean power [dB]', 'y')
[18]         ylim([-25,1])
[19]         xlim([0,0.4])
[20]         graphToPDF('../buoyPSDlog.pdf',4,3)
[21]     end

[1] function [pout, fout, Ns] = myPSD(data,fs,Nfft)
[2] % Computes the power spectral density of the data
[3] % which is recorded with the sample rate fs.
[4] % The data is cut into blocks of size Nfft.
[5] % It is assumed that the data is real so that the
[6] % negative spectrum is redundant and ignored.
[7] M = size(data,2);
[8]
[9] Nh = Nfft/2;
[10] L = length(data);
[11] Ns = floor(L/Nh)-1;
[12] %w = hamming(Nfft);
[13] w = hann(Nfft);
[14] pw = zeros(Nh,M);
[15] for m = 1:M
[16]     for n = 0:Ns-1
[17]         Noff = n*Nh;
[18]         H = fft(w.*detrend(data(Noff+(1:Nfft),m)));
[19]         pw(:,m) = pw(:,m) + abs(H(1:Nh)).^2;
[20]     end
[21] end
[22] df = fs/Nfft;
[23]
[24] fout = df*((1:Nh)-1);
[25] pout = pw/(Ns*Nfft);
[26] % In the above the Ns is for the mean and the
[27] % Nfft is for the scaling of the FFT.
[28] end

```

**3.6** This code solves the problem.

```

[1] function backscatter(par,type)
[2] % The input parameter "type" is a character a,b,c,d,or e.
[3] % The input parameter "par" depends on the type, look at
[4] % the function makeSurf() to see what the "par" is.
[5]
[6] N      = 2000;    % the number of grid points (1-D)
[7] angle = 0.01;    % the antenna beam width from center
[8] h      = 717e3;  % altitude of satellite
[9] dr     = 0.5;    % range resolution.
[10]
[11] % lay out the spacial grid on the sea surface.
[12] x_max = 1.25*h*angle;          % Size of grid
[13] x = x_max * 2*((0:(N-1))/(N-1)-0.5); % The grid array
[14] y = x';
[15] rho2 = x.^2 + y.^2;          % Distance from the center squared.
[16]
[17] alpha = 0.5/(h*angle)^2;      % A temporary constant
[18] G = exp(-alpha*rho2);        % The antenna gain
[19]
[20] z = makeSurf(x,y,par,type);   % Compute the surface height.
[21] r = sqrt((h-z).^2 + rho2);    % Range to satellite.
[22] d = r-h;                     % Subtract altitude.
[23]
[24] d_min = -20*dr;              % Location of first bin.
[25]
[26] Np = 128;
[27] dgrid = d_min+(0:(Np-1))*dr; % Create range bins.
[28]
[29] P = NaN(1,Np);              % Allocate the power array.
[30]
[31] for n = 1:Np                 % Loop over range bins.
[32]     d_target = dgrid(n);     % The range bin in this round.
[33]                                     % The next line does the magic.
[34]     inBin = abs(d-d_target)<(dr/2); % Array of 1's and 0's.
[35]     P(n) = sum(sum( G.*inBin )); % Power in bin n.
[36] end
[37]
[38] figure(2)
[39] plot(dgrid,P,'.')           % plot the power versus range
[40] ylim([0,1.1*max(P)])      % set the range of the plot
[41]
[42] end
[43]
[44] % This function creates the surface height array.
[45] function z = makeSurf(x,y,par,type)
[46]     H = par(1);
[47]     Nx = size(x,2);
[48]     Ny = size(y,1);
[49]     switch type
[50]         case 'a'
[51]             z = 0;
[52]         case 'b'
[53]             T = par(2);

```

```
[54]         theta = par(3);
[55]         k = 4*pi^2/(9.8*T^2);
[56]         kx = k*cosd(theta);
[57]         ky = k*sind(theta);
[58]         z = 0.5*H*cos(kx*x +ky*y);
[59]     case 'c'
[60]         z = H*(rand(Ny,Nx)-0.5);
[61]     case 'd'
[62]         z = 0.25*H*randn(Ny,Nx);
[63]     case 'e'
[64]         Hs = par(2);
[65]         T = par(3);
[66]         theta = par(4);
[67]         k = 4*pi^2/(9.8*T^2);
[68]         kx = k*cosd(theta);
[69]         ky = k*sind(theta);
[70]         z = 0.5*Hs*cos(kx*x +ky*y);
[71]         figure(10)
[72]         imagesc(z)
[73]         z = z + 0.25*H*randn(Ny,Nx);
[74]     end
[75]     figure(11)
[76]     imagesc(z)
[77] end
[78]
```

4.1 First consider a bug in the water at a distance  $a$  from the center of the bucket. The position of the bug will be

$$\vec{r} = a \cos \omega t \hat{x} + a \sin \omega t \hat{y}$$

This implies that

$$\begin{aligned} \vec{v} &= \frac{d\vec{r}}{dt} = -\omega a \sin \omega t \hat{x} + \omega a \cos \omega t \hat{y} + 0 \hat{z} \\ &= -\omega r_y \hat{x} + \omega r_x \hat{y} \end{aligned}$$

Thus when the bug is at the position  $(x, y)$  it has velocity

$$\vec{v} = -\omega y \hat{x} + \omega x \hat{y}$$

and thus since  $\vec{v}$  and  $\vec{u}$  are the same when they are in the same place

$$\vec{u} = -\omega y \hat{x} + \omega x \hat{y}$$

Now we can compute.

$$\vec{u} \cdot \nabla = (-\omega y \hat{x} + \omega x \hat{y}) \cdot \left( \frac{\partial}{\partial x} \hat{x} + \frac{\partial}{\partial y} \hat{y} + \frac{\partial}{\partial z} \hat{z} \right) = -\omega y \frac{\partial}{\partial x} + \omega x \frac{\partial}{\partial y}$$

and

$$\begin{aligned} (\vec{u} \cdot \nabla) \vec{u} &= (-\omega y \frac{\partial}{\partial x} + \omega x \frac{\partial}{\partial y}) \vec{u} \\ &= -\omega y \frac{\partial \vec{u}}{\partial x} + \omega x \frac{\partial \vec{u}}{\partial y} \\ &= -\omega y \frac{\partial(-\omega y \hat{x} + \omega x \hat{y})}{\partial x} + \omega x \frac{\partial(-\omega y \hat{x} + \omega x \hat{y})}{\partial y} \\ &= -\omega y \frac{\partial(\omega x \hat{y})}{\partial x} + \omega x \frac{\partial(-\omega y \hat{x})}{\partial y} \\ &= -\omega^2 y \hat{y} - \omega^2 x \hat{x} \\ &= -\omega^2 (x \hat{x} + y \hat{y}) \\ &= -\omega^2 \vec{r} \end{aligned}$$

and so

$$\vec{a} = \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = 0 - \omega^2 \vec{r} = -\omega^2 \vec{r}$$

This is pointed toward the center as expected and since  $\omega = \frac{v}{r}$  the magnitude is  $\omega^2 r = \frac{v^2}{r^2} r = \frac{v^2}{r}$  which is also what we expect for circular motion. Now we can compute the curl.

$$\begin{aligned} \nabla \times \vec{u} &= \left( \frac{\partial}{\partial x} \hat{x} + \frac{\partial}{\partial y} \hat{y} + \frac{\partial}{\partial z} \hat{z} \right) \times (-\omega y \hat{x} + \omega x \hat{y}) \\ &= \left( \frac{\partial}{\partial x} \hat{x} + \frac{\partial}{\partial y} \hat{y} \right) \times (-\omega y \hat{x} + \omega x \hat{y}) \\ &= \frac{\partial}{\partial x} \hat{x} \times (-\omega y \hat{x} + \omega x \hat{y}) + \frac{\partial}{\partial y} \hat{y} \times (-\omega y \hat{x} + \omega x \hat{y}) \\ &= \frac{\partial}{\partial x} \hat{x} \times (0 + \omega x \hat{y}) + \frac{\partial}{\partial y} \hat{y} \times (-\omega y \hat{x} + 0) \\ &= \frac{\partial}{\partial x} \omega x \hat{z} + \frac{\partial}{\partial y} \omega y \hat{z} \\ &= \omega \hat{z} + \omega \hat{z} \\ &= 2\omega \hat{z} \end{aligned}$$

So that

$$\begin{aligned}
 \vec{u} \times (\nabla \times \vec{u}) &= 2\omega \vec{u} \times \hat{z} \\
 &= 2\omega (-\omega y \hat{x} + \omega x \hat{y}) \times \hat{z} \\
 &= 2\omega (\omega y \hat{y} + \omega x \hat{x}) \\
 &= 2\omega^2 (y \hat{y} + x \hat{x}) \\
 &= 2\omega^2 \vec{r}
 \end{aligned}$$

Thus the RHS of  $\frac{1}{2}\nabla(\vec{u} \cdot \vec{u}) = (\vec{u} \cdot \nabla)\vec{u} + \vec{u} \times (\nabla \times \vec{u})$  is

$$(\vec{u} \cdot \nabla)\vec{u} + \vec{u} \times (\nabla \times \vec{u}) = -\omega^2 \vec{r} + 2\omega^2 \vec{r} = \omega^2 \vec{r}$$

Now for the LHS

$$\begin{aligned}
 \frac{1}{2}\nabla(\vec{u} \cdot \vec{u}) &= \frac{1}{2}\nabla(\omega^2 y^2 + \omega^2 x^2) \\
 &= \frac{1}{2} \left( \frac{\partial}{\partial x} \hat{x} + \frac{\partial}{\partial y} \hat{y} + \frac{\partial}{\partial z} \hat{z} \right) (\omega^2 y^2 + \omega^2 x^2) \\
 &= \frac{1}{2} \left( \frac{\partial}{\partial x} \hat{x} + \frac{\partial}{\partial y} \hat{y} \right) (\omega^2 y^2 + \omega^2 x^2) \\
 &= \frac{1}{2} (2\omega^2 x \hat{x} + 2\omega^2 y \hat{y}) \\
 &= \omega^2 \vec{r}
 \end{aligned}$$

which is the same as the RHS!!!!!!

4.2 Starting the the LHS.

$$\begin{aligned}\nabla \cdot [\rho \vec{u}] &= \partial_i [\rho \vec{u}]_i \\ &= u_i \partial_i \rho + \rho \partial_i u_i \\ &= u_i \partial_i \rho + \rho \partial_i u_i \\ &= \vec{u} \cdot \nabla \rho + \rho \nabla \cdot \vec{u}\end{aligned}$$

4.3 Let's start with writing out the  $k$ -th component of the RHS.

$$\begin{aligned}
 \left[ \frac{1}{2} \nabla(\vec{u} \cdot \vec{u}) - (\vec{u} \cdot \nabla) \vec{u} \right]_k &= \frac{1}{2} \partial_k (\vec{u} \cdot \vec{u}) - (\vec{u} \cdot \nabla) u_k \\
 &= \frac{1}{2} \partial_k (u_i u_i) - (u_i \partial_i) u_k \\
 &= u_i \partial_k u_i - u_i \partial_i u_k \\
 &= \sum_i [u_i \partial_k u_i - u_i \partial_i u_k] \\
 &= \sum_{i \neq k} [u_i \partial_k u_i - u_i \partial_i u_k] + [u_k \partial_k u_k - u_k \partial_k u_k] \\
 &= \sum_{i \neq k} [u_i \partial_k u_i - u_i \partial_i u_k]
 \end{aligned}$$

Now let us work on the  $k$ -th component of the LHS.

$$\begin{aligned}
 [\vec{u} \times (\nabla \times \vec{u})]_k &= \varepsilon_{ijk} u_i (\nabla \times \vec{u})_j \\
 &= \varepsilon_{ijk} u_i \varepsilon_{nmj} \partial_n u_m \\
 &= \varepsilon_{ijk} \varepsilon_{nmj} u_i \partial_n u_m \\
 &= -\varepsilon_{ijk} \varepsilon_{njm} u_i \partial_n u_m
 \end{aligned}$$

In this expression there is an implicit sum over  $n, m, i$  and  $j$  but as always most of them give zero. The only terms that we need to be concerned with are those for which both  $\varepsilon_{ijk} \neq 0$  and  $\varepsilon_{nmj} \neq 0$ . In particular in the sum over  $n$  and  $m$  the only terms that are not zero is when  $(n, m) = (k, i)$  and when  $(n, m) = (i, k)$ , since there is a common  $j$  the two Levi-Civita symbols. Thus

$$\begin{aligned}
 [\vec{u} \times (\nabla \times \vec{u})]_k &= -\varepsilon_{ijk} \varepsilon_{kji} u_i \partial_k u_i - \varepsilon_{ijk} \varepsilon_{ijk} u_i \partial_i u_k \\
 &= \varepsilon_{ijk} \varepsilon_{ijk} u_i \partial_k u_i - \varepsilon_{ijk} \varepsilon_{ijk} u_i \partial_i u_k \\
 &= \varepsilon_{ijk} \varepsilon_{ijk} [u_i \partial_k u_i - u_i \partial_i u_k] \\
 &= \sum_i^{\neq k} \sum_j^{\neq k \text{ and } \neq i} \varepsilon_{ijk} \varepsilon_{ijk} [u_i \partial_k u_i - u_i \partial_i u_k] \\
 &= \sum_i^{\neq k} [u_i \partial_k u_i - u_i \partial_i u_k]
 \end{aligned}$$

So we see that the LHS and RHS are the same.

4.4 Start with the LHS.

$$\begin{aligned}
[\nabla \times (\vec{u} \times \vec{v})]_k &= \varepsilon_{ijk} \partial_i [\vec{u} \times \vec{v}]_j \\
&= \varepsilon_{ijk} \partial_i [\varepsilon_{nmj} u_n v_m] \\
&= \varepsilon_{ijk} \varepsilon_{nmj} \partial_i [u_n v_m] \\
&= -\varepsilon_{ijk} \varepsilon_{njm} \partial_i [u_n v_m] \\
&= -\varepsilon_{ijk} \varepsilon_{njm} [v_m \partial_i u_n + u_n \partial_i v_m] \\
&= -\varepsilon_{ijk} \varepsilon_{kji} [v_i \partial_i u_k + u_k \partial_i v_i] - \varepsilon_{ijk} \varepsilon_{ijk} [v_k \partial_i u_i + u_i \partial_i v_k] \\
&= \varepsilon_{ijk} \varepsilon_{ijk} [v_i \partial_i u_k + u_k \partial_i v_i - v_k \partial_i u_i - u_i \partial_i v_k] \\
&= \sum_{i \neq k} [v_i \partial_i u_k + u_k \partial_i v_i - v_k \partial_i u_i - u_i \partial_i v_k] \\
&= \sum_{i \neq k} [v_i \partial_i u_k + u_k \partial_i v_i - v_k \partial_i u_i - u_i \partial_i v_k] + 0 \\
&= \sum_{i \neq k} [v_i \partial_i u_k + u_k \partial_i v_i - v_k \partial_i u_i - u_i \partial_i v_k] \\
&\quad + v_k \partial_k u_k + u_k \partial_k v_k - v_k \partial_k u_k - u_k \partial_k v_k \\
&= \sum_i [v_i \partial_i u_k + u_k \partial_i v_i - v_k \partial_i u_i - u_i \partial_i v_k] \\
&= v_i \partial_i u_k + u_k \partial_i v_i - v_k \partial_i u_i - u_i \partial_i v_k \\
&= v_i \partial_i u_k + u_k \partial_i v_i - v_k \partial_i u_i - u_i \partial_i v_k \\
&= u_k \partial_i v_i - v_k \partial_i u_i + v_i \partial_i u_k - u_i \partial_i v_k
\end{aligned}$$

Thus

$$\begin{aligned}
\nabla \times (\vec{u} \times \vec{v}) &= [\nabla \times (\vec{u} \times \vec{v})]_k \hat{e}_k \\
&= (u_k \partial_i v_i - v_k \partial_i u_i + v_i \partial_i u_k - u_i \partial_i v_k) \hat{e}_k \\
&= u_k \hat{e}_k \partial_i v_i - v_k \hat{e}_k \partial_i u_i + v_i \partial_i u_k \hat{e}_k - u_i \partial_i v_k \hat{e}_k \\
&= \vec{u} (\nabla \cdot \vec{v}) - \vec{v} (\nabla \cdot \vec{u}) + (\vec{v} \cdot \nabla) \vec{u} - (\vec{u} \cdot \nabla) \vec{v}
\end{aligned}$$



4.5 First we compute the divergence.

(a)

$$\nabla \cdot \vec{u} = \nabla \cdot (-\omega y \hat{x} + \omega x \hat{y}) = \frac{\partial}{\partial x}(-\omega y) + \frac{\partial}{\partial y}(\omega x) = 0 + 0 = 0$$

So this satisfies the equation of continuity for an incompressible fluid.

(b)

$$\begin{aligned} \nabla \cdot \vec{u} &= \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} \\ &= \omega + (-\omega) \\ &= 0 \end{aligned}$$

So this satisfies the equation of continuity for an incompressible fluid.

(c)

$$\begin{aligned} \nabla \cdot \vec{u} &= \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} \\ &= -2\alpha x + 2\alpha x \\ &= 0 \end{aligned}$$

So this satisfies the equation of continuity for an incompressible fluid.

(d)

$$\begin{aligned} \nabla \cdot \vec{u} &= \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} \\ &= \cosh(\kappa x) \cosh(\kappa x) - \cosh(\kappa x) \cosh(\kappa x) \\ &= 0 \end{aligned}$$

So this satisfies the equation of continuity for an incompressible fluid.

**4.6** Cylindrical coordinates  $(r, \theta, z)$  will be helpful here. There is nothing special about any particular angle  $\theta$ , so we expect the pressure to depend only on  $r$  and  $z$ , and not  $\theta$ . Thus

$$\begin{aligned}\nabla P &= \frac{\partial P}{\partial r} \hat{r} + \frac{1}{r} \frac{\partial P}{\partial \theta} \hat{\theta} + \frac{\partial P}{\partial z} \hat{z} \\ &= \frac{\partial P}{\partial r} \hat{r} + \frac{\partial P}{\partial z} \hat{z}\end{aligned}$$

Thus the equation of motion

$$\frac{\partial \vec{u}}{\partial t} + \frac{1}{2} \nabla(u^2) - \vec{u} \times (\nabla \times \vec{u}) + \frac{\nabla P}{\rho} + g \hat{z} = 0$$

becomes

$$-\omega^2 \vec{r} + \frac{1}{\rho} \left( \frac{\partial P}{\partial r} \hat{r} + \frac{\partial P}{\partial z} \hat{z} \right) + g \hat{z} = 0$$

If we look at the  $\hat{z}$  direction we find that

$$\frac{\partial P}{\partial z} = -\rho g \quad \longrightarrow \quad P = -\rho g z + f(r) + \text{constant}$$

while for the  $\hat{r}$  direction

$$\frac{\partial P}{\partial r} = \rho \omega^2 r \quad \longrightarrow \quad P = \frac{1}{2} \rho \omega^2 r^2 + h(z) + \text{constant}$$

Thus we see that

$$P = \frac{1}{2} \rho \omega^2 r^2 - \rho g z + \text{constant}$$

## 4.7

(a)

$$\vec{u} = -\omega y \hat{x} + \omega x \hat{y}$$

First we note that  $\frac{\partial}{\partial z} \vec{u} = 0$  and  $u_z = 0$  so that

$$\nabla \times \vec{u} = \left( \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \right) \hat{z} = (\omega - (-\omega)) \hat{z} = -2\omega \hat{z}$$

So there is no scalar field whose gradient is  $\vec{u}$ .

(b)

$$\vec{u} = \omega x \hat{x} - \omega y \hat{y}$$

$$\nabla \times \vec{u} = \left( \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \right) \hat{z} = (0 - 0) \hat{z} = 0$$

So yes we can write this as the gradient of a scalar field. Thus we have that

$$\frac{\partial \phi}{\partial x} = u_x = \omega x \longrightarrow \phi = \frac{1}{2} \omega x^2 + f(y)$$

$$\frac{\partial \phi}{\partial y} = u_y = -\omega y \longrightarrow \phi = -\frac{1}{2} \omega y^2 + h(x)$$

So it appears that the function

$$\phi = \frac{1}{2} \omega (x^2 - y^2)$$

does the trick.

(c) With

$$\vec{u} = \alpha(y^2 - x^2) \hat{x} + 2\alpha xy \hat{y}$$

$$\nabla \times \vec{u} = \left( \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \right) \hat{z} = (2\alpha y - 2\alpha y) \hat{z} = 0$$

So yes we can write this as the gradient of a scalar field. Thus we have that

$$\frac{\partial \phi}{\partial x} = u_x = \alpha(y^2 - x^2) \longrightarrow \phi = \alpha(xy^2 - \frac{1}{3}x^3) + f(y)$$

$$\frac{\partial \phi}{\partial y} = u_y = 2\alpha xy \longrightarrow \phi = \alpha xy^2 + h(x)$$

So the function

$$\phi = \alpha(xy^2 - \frac{1}{3}x^3)$$

does the trick.

(d)

$$\vec{u} = \sinh(\kappa x) \cosh(\kappa y) \hat{x} - \cosh(\kappa x) \sinh(\kappa y) \hat{y}$$

$$\begin{aligned} \nabla \times \vec{u} &= \left( \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \right) \hat{z} \\ &= (-\sinh(\kappa x) \sinh(\kappa y) - \sinh(\kappa x) \sinh(\kappa y)) \hat{z} \\ &= -2 \sinh(\kappa x) \sinh(\kappa y) \hat{z} \end{aligned}$$

So there is no scalar field whose gradient is  $\vec{u}$ .

## 4.8

(a) The BC at the bottom is

$$\begin{aligned}\hat{n} \cdot \nabla \phi \Big|_{z=-d} &= \hat{z} \cdot \nabla \phi \Big|_{z=-d} = \frac{\partial \phi}{\partial z} \Big|_{z=-d} \\ &= kA \cos(\omega t) \sinh(k(z+d)) \cos(kx - D) \Big|_{z=-d} \\ &= kA \cos(\omega t) \sinh(0) \cos(kx - D) \\ &= 0\end{aligned}$$

So the BC on the bottom is ok.

(b) First the condition at  $x = 0$ .

$$\begin{aligned}0 &= \hat{n} \cdot \nabla \phi \Big|_{x=0} = \hat{x} \cdot \nabla \phi \Big|_{x=0} = \frac{\partial \phi}{\partial x} \Big|_{x=0} \\ &= -kA \cos(\omega t) \cosh(k(z+d)) \sin(kx - D) \Big|_{x=0} \\ &= -kA \cos(\omega t) \cosh(k(z+d)) \sin(-D)\end{aligned}$$

This must be true for all  $t$  and  $z$  so we need that  $\sin(-D) = 0$  which means that  $D = n\pi$ . We choose  $D = 0$  without loss of generality since the constant  $A$  can be positive or negative. So now have

$$\phi = A \cos(\omega t) \cosh(k(z+d)) \cos(kx)$$

Now for the condition at  $x = L$

$$\begin{aligned}0 &= \hat{n} \cdot \nabla \phi \Big|_{x=L} = \hat{x} \cdot \nabla \phi \Big|_{x=L} = \frac{\partial \phi}{\partial x} \Big|_{x=L} \\ &= -kA \cos(\omega t) \cosh(k(z+d)) \sin(kx) \Big|_{x=L} \\ &= -kA \cos(\omega t) \cosh(k(z+d)) \sin(kL) \\ &\rightarrow \sin(kL) = 0 \\ &\rightarrow kL = n\pi \\ &\rightarrow k = \frac{n\pi}{L}\end{aligned}$$

So  $k$  must be a multiple of  $\pi/L$ .

(c) We start with the Dynamic BC

$$\begin{aligned}\left[ \frac{\partial \phi}{\partial t} \right]_{z=0} + g\eta &= 0 \\ \rightarrow \eta &= -\frac{1}{g} \left[ \frac{\partial \phi}{\partial t} \right]_{z=0} \\ &= -\frac{1}{g} [-\omega A \sin(\omega t) \cosh(k(z+d)) \cos(kx)]_{z=0} \\ &= \frac{\omega A}{g} \sin(\omega t) \cosh(kd) \cos(kx) \\ \rightarrow \frac{\partial \eta}{\partial t} &= \frac{\omega^2 A}{g} \cos(\omega t) \cosh(kd) \cos(kx)\end{aligned}$$

(d) Now we check the kinematic BC.

$$\begin{aligned}\left[\frac{\partial\phi}{\partial z}\right]_{z=0} - \frac{\partial\eta}{\partial t} &= 0 \\ \rightarrow \left[\frac{\partial\phi}{\partial z}\right]_{z=0} &= \frac{\partial\eta}{\partial t} \\ \rightarrow kA \cos(\omega t) \sinh(kd) \cos(kx) &= \frac{\omega^2 A}{g} \cos(\omega t) \cosh(kd) \cos(kx) \\ \rightarrow k \sinh(kd) &= \frac{\omega^2}{g} \cosh(kd) \\ \rightarrow gk \tanh(kd) &= \omega^2\end{aligned}$$

So we see that there is a direct relationship between  $k$  and  $\omega$  that also depends on the depth  $d$ .

4.9 We use the definition of  $\phi$  and compute.

$$\nabla^2 \phi = \nabla^2(a_1 \phi_1 + a_2 \phi_2) = a_1 \nabla^2 \phi_1 + a_2 \nabla^2 \phi_2 = a_1 \cdot 0 + a_2 \cdot 0 = 0$$

In a similar way

$$\begin{aligned} \left[ \frac{\partial \phi}{\partial z} \right]_{z=-d} &= \left[ \frac{\partial}{\partial z} (a_1 \phi_1 + a_2 \phi_2) \right]_{z=-d} \\ &= \left[ a_1 \frac{\partial \phi_1}{\partial z} + a_2 \frac{\partial \phi_2}{\partial z} \right]_{z=-d} \\ &= a_1 \cdot 0 + a_2 \cdot 0 \\ &= 0 \end{aligned}$$

4.10 With  $\phi = E \cosh \zeta \cos \xi$ .

(a)

$$\begin{aligned}\nabla\phi &= \frac{\partial\phi}{\partial x}\hat{x} + \frac{\partial\phi}{\partial z}\hat{z} \\ &= k\frac{\partial\phi}{\partial\xi}\hat{x} + k\frac{\partial\phi}{\partial\zeta}\hat{z} \\ &= -kE\cosh\zeta\sin\xi\hat{x} + kE\sinh\zeta\cos\xi\hat{z}\end{aligned}$$

(b)

$$\begin{aligned}|\nabla\phi|^2 &= \nabla\phi \cdot \nabla\phi \\ &= k^2E^2(\cosh^2\zeta\sin^2\xi + \sinh^2\zeta\cos^2\xi) \\ &= k^2E^2((\sinh^2\zeta + 1)\sin^2\xi + \sinh^2\zeta\cos^2\xi) \\ &= k^2E^2(\sinh^2\zeta + \sin^2\xi)\end{aligned}$$

(c)

$$\begin{aligned}\frac{\partial\phi}{\partial t} &= \frac{\partial}{\partial t}[E\cosh\zeta\cos\xi] \\ &= \frac{dE}{dt}\cosh\zeta\cos\xi + E\cosh\zeta\frac{\partial}{\partial t}[\cos\xi] \\ &= \frac{dE}{dt}\cosh\zeta\cos\xi - E\cosh\zeta\sin\xi\frac{\partial\xi}{\partial t} \\ &= \frac{dE}{dt}\cosh\zeta\cos\xi + E\cosh\zeta\sin\xi\frac{dD}{dt} \\ &= \cosh\zeta\left(\frac{dE}{dt}\cos\xi + E\frac{dD}{dt}\sin\xi\right)\end{aligned}$$

4.11 We start with the Dynamic BC

$$\begin{aligned} \left[ \frac{\partial \phi}{\partial t} \right]_{z=0} + g\eta &= 0 \\ \longrightarrow \eta &= -\frac{1}{g} \left[ \frac{\partial \phi}{\partial t} \right]_{z=0} \\ &= -\frac{1}{g} [\omega E \cosh \zeta \sin \xi]_{z=0} \\ &= -\frac{\omega E}{g} \cosh(kd) \sin \xi \\ \longrightarrow \frac{\partial \eta}{\partial t} &= \frac{\omega^2 E}{g} \cosh(kd) \cos \xi \end{aligned}$$

Now we check the kinematic BC.

$$\begin{aligned} \left[ \frac{\partial \phi}{\partial z} \right]_{z=0} - \frac{\partial \eta}{\partial t} &= 0 \\ \longrightarrow \left[ \frac{\partial \phi}{\partial z} \right]_{z=0} &= \frac{\partial \eta}{\partial t} \\ \longrightarrow kE \sinh(kd) \cos \xi &= \frac{\omega^2 E}{g} \cosh(kd) \cos \xi \\ \longrightarrow k \sinh(kd) &= \frac{\omega^2}{g} \cosh(kd) \\ \longrightarrow gk \tanh(kd) &= \omega^2 \end{aligned}$$

So we see that there is a direct relationship between  $k$  and  $\omega$  that also depends on the depth  $d$ .



## 4.12

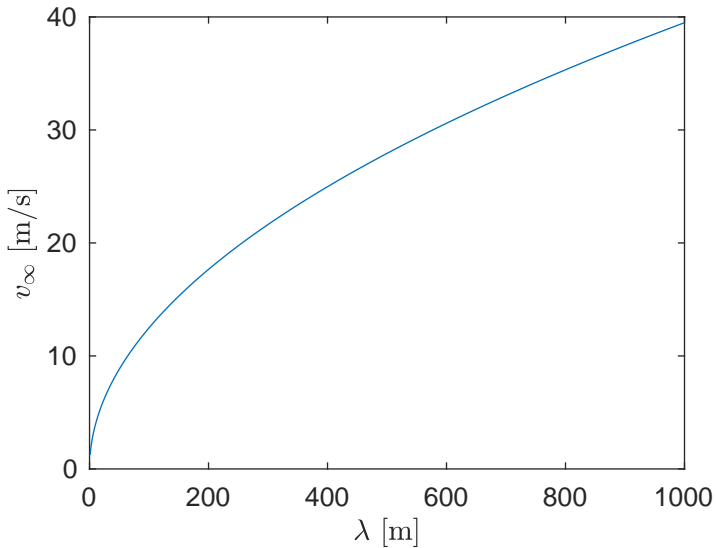
$$\omega^2 = gk \tanh(kd) \rightarrow \frac{\omega^2}{k^2} = \frac{g \tanh(kd)}{k} \rightarrow v_p = \frac{\omega}{k} = \sqrt{\frac{g \tanh(kd)}{k}}$$

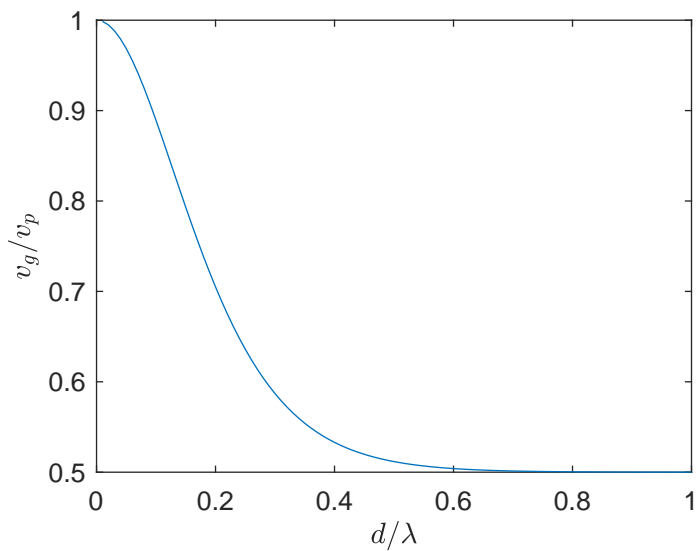
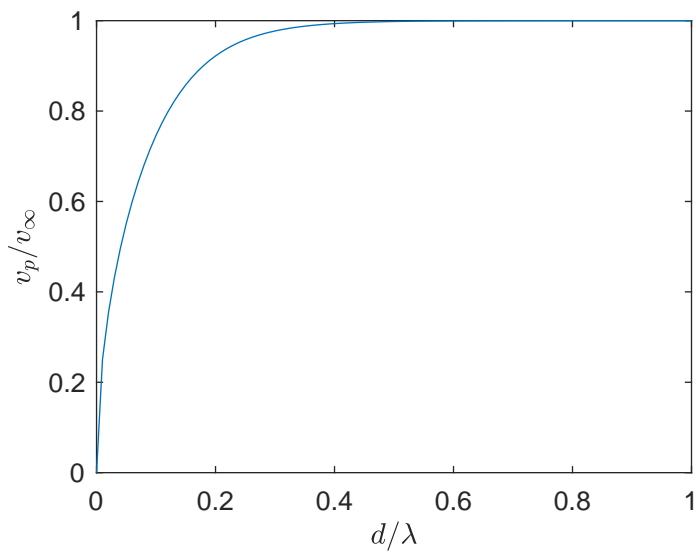
This

$$v_\infty = \lim_{d \rightarrow \infty} v_p = \sqrt{\frac{g}{k}}$$

While

$$\begin{aligned} \omega^2 &= gk \tanh(kd) \\ \rightarrow \frac{\partial}{\partial k} [\omega^2] &= \frac{\partial}{\partial k} [gk \tanh(kd)] \\ \rightarrow 2\omega \frac{\partial \omega}{\partial k} &= g \tanh(kd) + \frac{gkd}{\cosh^2(kd)} \\ \rightarrow v_g = \frac{\partial \omega}{\partial k} &= \frac{g \tanh(kd)}{2\omega} + \frac{gkd}{2\omega \cosh^2(kd)} \\ \frac{v_g}{v_p} &= \frac{gk \tanh(kd)}{2\omega^2} + \frac{gk d}{2\omega^2 \cosh^2(kd)} \\ &= \frac{1}{2} + \frac{kd}{2 \cosh^2(kd) \tanh(kd)} \\ &= \frac{1}{2} + \frac{kd}{2 \cosh(kd) \sinh(kd)} \\ &= \frac{1}{2} + \frac{kd}{\sinh(2kd)} \end{aligned}$$

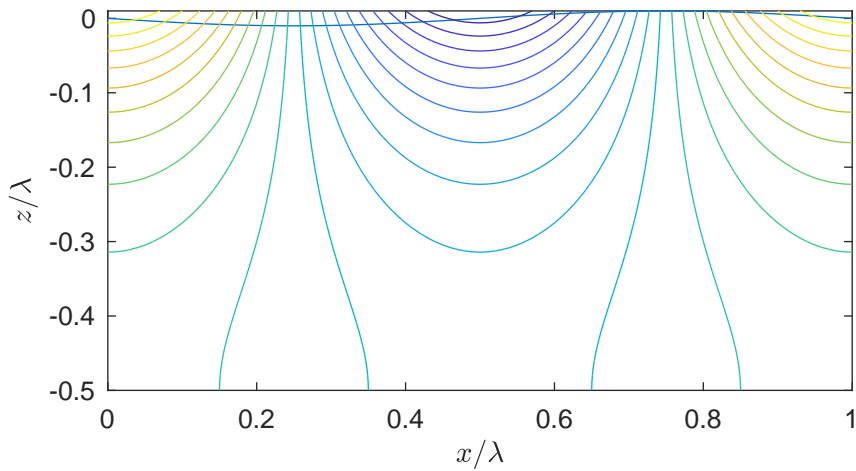
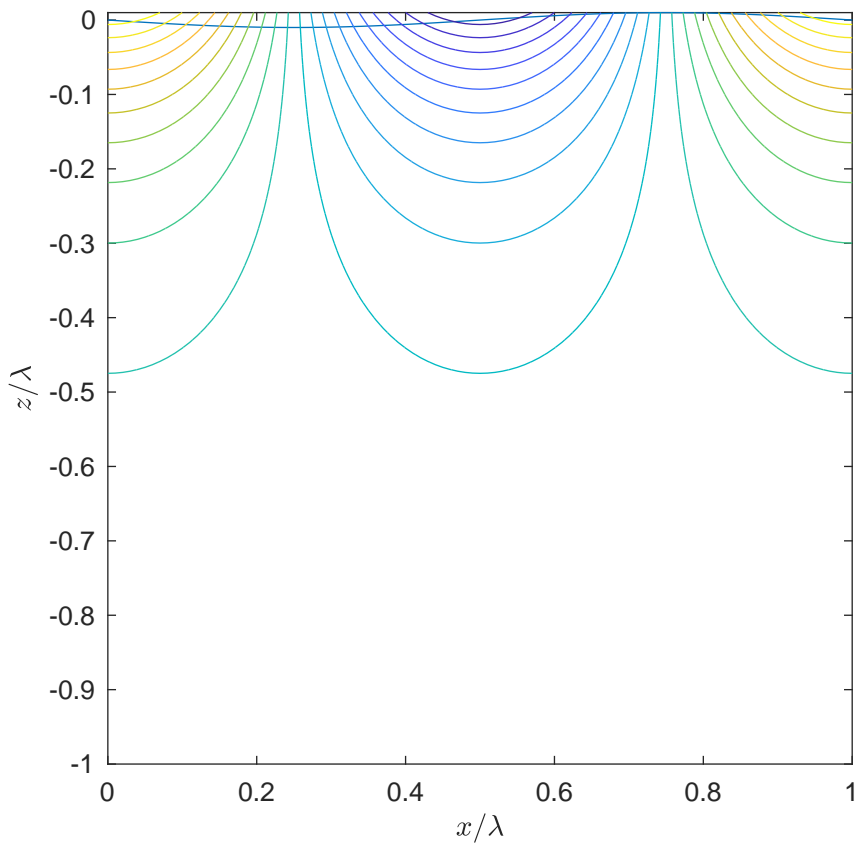


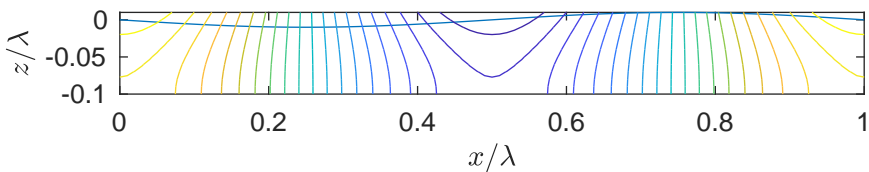
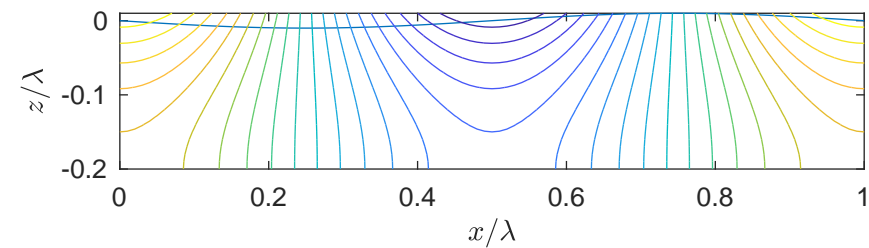


4.13 Here is the code that makes the following graphs.

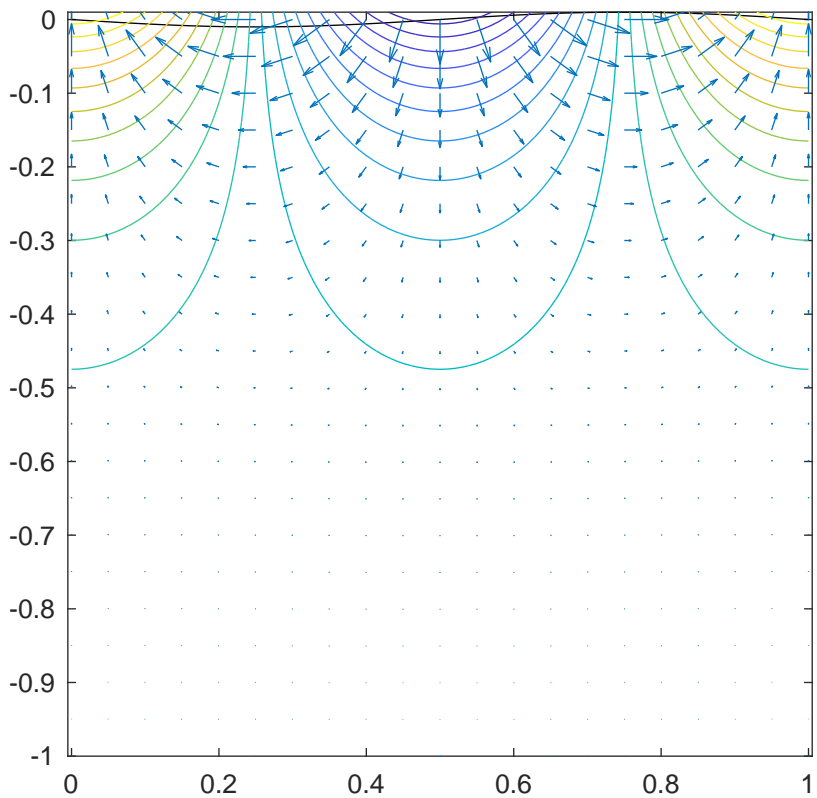
```
[1] function makeGraph2(d0lambda)
[2] dx    = 0.01;
[3] z0lam = (-d0lambda:dx:dx);
[4] x0lam = (0:dx:1);
[5] xi    = 2*pi*x0lam;
[6] zeta  = 2*pi*(z0lam+d0lambda)';
[7] phi   = cosh(zeta)*cos(xi);
[8]
[9] figure(1) % the contour graph
[10] plot(x0lam,-dx*sin(xi)) % plot surface wave
[11] hold on
[12] contour(x0lam,z0lam,phi,20) % plot contour
[13] texText('$z/\lambda$', 'y') % add lables
[14] texText('$x/\lambda$', 'x')
[15] axis equal
[16] hold off
[17] name = sprintf('contour%03d.pdf',100*d0lambda);
[18] graphToPDF(name,1+4,1+4*d0lambda)
[19]
[20] figure(2) % the contour graph with velocity field.
[21] dx    = 0.05;
[22] plot(x0lam,-0.01*sin(xi),'k') % plot surface wave
[23] hold on
[24] contour(x0lam,z0lam,phi,20) % plot contour
[25]
[26] z0lam = (-d0lambda:dx:0); % compute velocity field
[27] x0lam = (0:dx:1);
[28] xi    = 2*pi*x0lam;
[29] zeta  = 2*pi*(z0lam+d0lambda)';
[30] ux = -cosh(zeta)*sin(xi);
[31] uz = sinh(zeta)*cos(xi);
[32] quiver(x0lam,z0lam,ux,uz) % plot the velocity field
[33] ylim([-d0lambda,0.01])
[34] axis equal
[35] hold off
[36] name = sprintf('contourQuiver%03d.pdf',100*d0lambda);
[37] graphToPDF(name,1+4,1+4*d0lambda)
[38] end
```

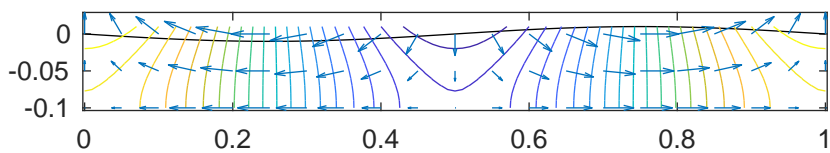
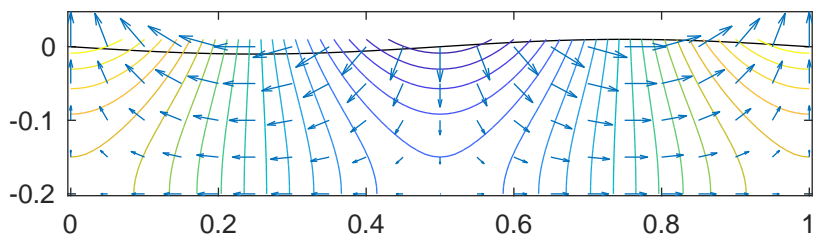
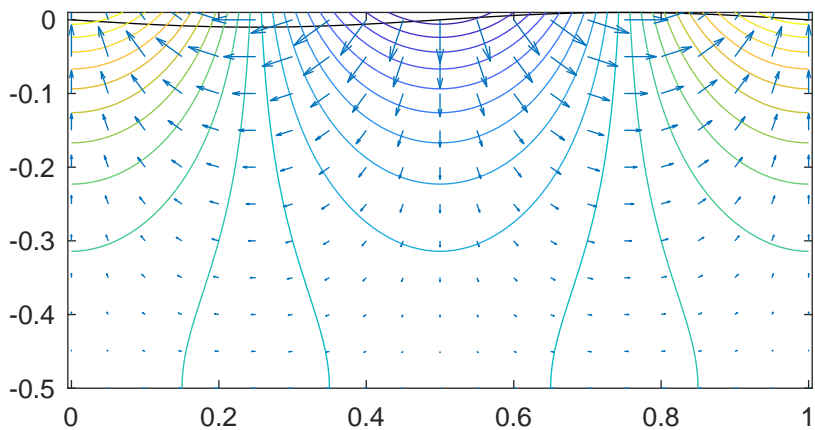
Here are the contours. Blue is low and yellow is high.





Here are the contours with the velocity field overlay.

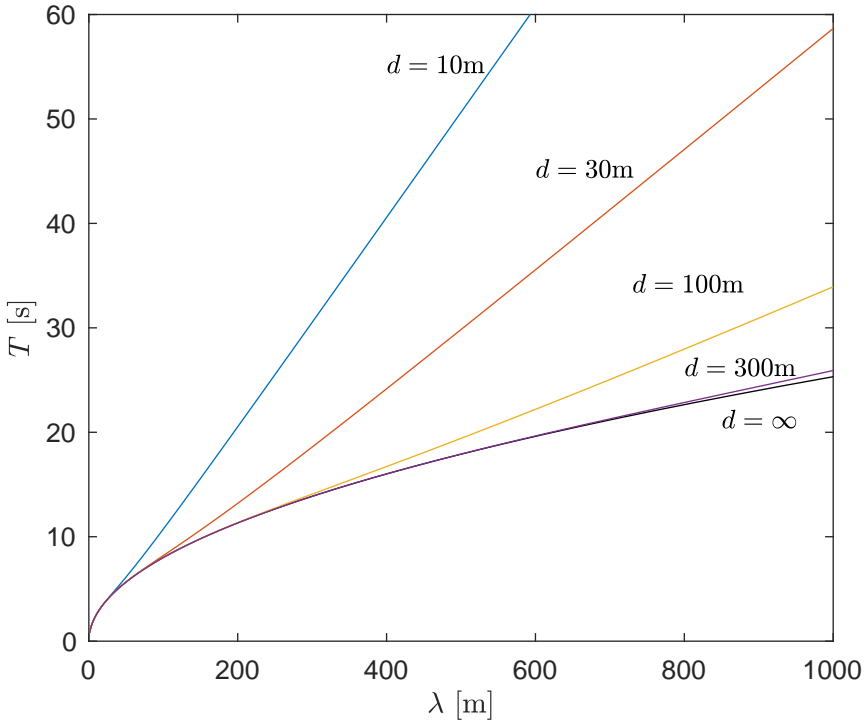




4.14 Taking the limits we find the follow.

$$\lim_{kd \rightarrow 0} v_p = \lim_{kd \rightarrow 0} \sqrt{\frac{g \tanh(kd)}{k}} = \sqrt{\frac{gkd}{k}} = \sqrt{gd}$$

$$\lim_{kd \rightarrow 0} \frac{v_g}{v_p} = \lim_{kd \rightarrow 0} \left[ \frac{1}{2} + \frac{kd}{\sinh(2kd)} \right] = \left[ \frac{1}{2} + \frac{kd}{2kd} \right] = 1$$



This code is what made the graph.

```
[1] function makeGraph3()
[2] d = [10,30,100,300];
[3] g = 9.8;
[4] N = 100;
[5] lam0 = 1;
[6] lamN = 1000;
[7] lam = lam0*exp((log(lamN/lam0)/N)*(0:N))';
[8] k = 2*pi./lam;
[9] omega = NaN(N+1,length(d));
[10]
[11] for n = 1:length(d)
[12]     omega(:,n) = sqrt(g*k.*tanh(k*d(n)));
[13] end
[14] T = 2*pi./omega;
[15] T_inf = 2*pi./sqrt(g*k);
[16]
[17] plot(lam,T_inf,'k')
[18] hold on
[19] plot(lam,T)
[20] hold off
[21]
[22] texText('$\lambda$ [m]','x')
[23] texText('$T$ [s]','y')
```

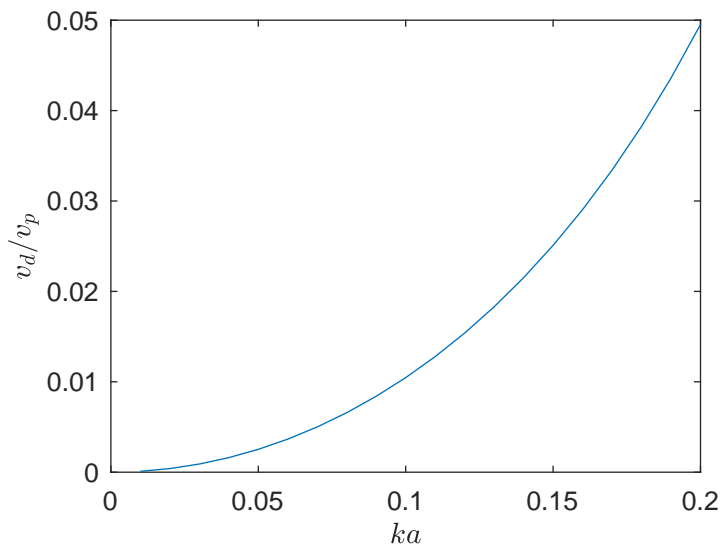
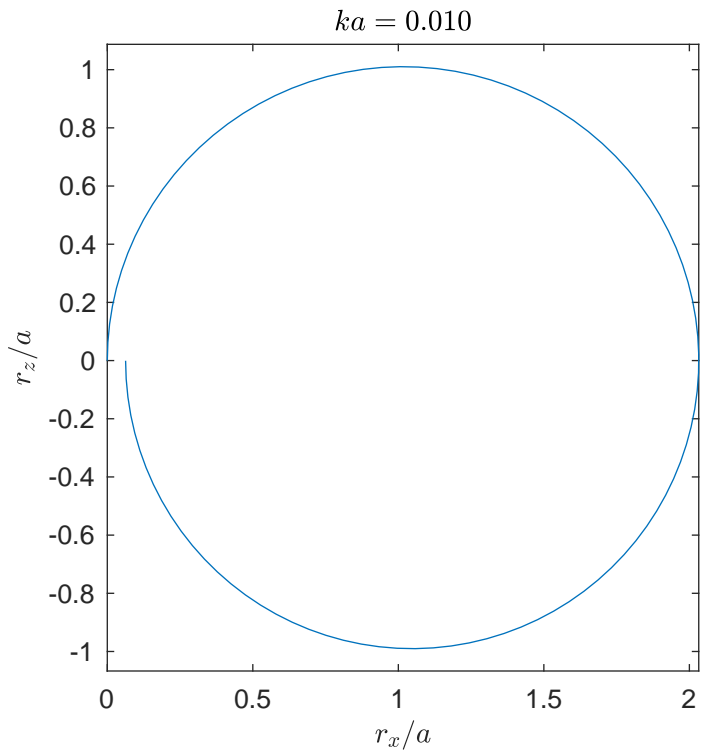
```
[24] ylim([0,60])
[25] texText('$d=10$m', 'g', 400, 55)
[26] texText('$d=30$m', 'g', 600, 45)
[27] texText('$d=100$m', 'g', 730, 34)
[28] texText('$d=300$m', 'g', 800, 26)
[29] texText('$d=\infty$', 'g', 850, 21)
[30] graphToPDF('period.pdf', 5, 4)
[31] end
```



4.15 This code solves the problem using the second order Runge-Kutta method.

```
[1] function velRatio = integrateRG2(ka,N,beta0)
[2] beta = NaN(N+1,2);
[3] beta(1,:) = beta0;
[4] dt = 2*pi/N;
[5] t = (0:N)*dt;
[6] for n = 1:N
[7]     betaMid    = beta(n,:) +vel(ka*beta(n,:),t(n))*dt/2;
[8]     tMid      = t(n)+dt/2;
[9]     beta(n+1,:)= beta(n,:) +vel(ka*betaMid,tMid)*dt;
[10] end
[11] figure(1)
[12] plot(beta(:,1),beta(:,2))
[13] axis equal
[14] texText('$r_x/a$', 'x')
[15] texText('$r_z/a$', 'y')
[16] texText(sprintf('$ka = %0.3f$',ka),'t')
[17] dBeta = beta(end,1)-beta(1,1);
[18] velRatio = ka*dBeta/(2*pi);
[19] end
[20]
[21] function u = vel(ab,t)
[22]     x = ab(1)-t;
[23]     z = ab(2);
[24]     u = exp(z)*[-sin(x),cos(x)];
[25] end
```

```
[1] function graphVelRatio(ka)
[2] N = length(ka);
[3] velRatio = NaN(1,N);
[4]
[5] for n = 1:N
[6]     velRatio(n) = integrateRG2(ka(n),100,[0,0]);
[7] end
[8] graphToPDF('trajectory.pdf',4,4);
[9] figure(10)
[10] plot(ka,velRatio)
[11] texText('$ka$', 'x')
[12] texText('$v_d/v_p$', 'y')
[13] graphToPDF('velRatio.pdf',4,3)
[14] end
```



**4.16** In this case  $d > \lambda$  so that  $2kd > 4\pi$  and  $e^{-2kd} < 0.0000035$  so we can use the approximation that  $\frac{\cosh(kz+kd)}{\sinh(kd)} \approx e^{kz}$  and  $\frac{\sinh(kz+kd)}{\sinh(kd)} \approx e^{kz}$  and

$$\vec{u} = \nabla\phi = ka v_p e^{kz} [-\sin\xi\hat{x} + \cos\xi\hat{z}]$$

(a) The trough is where  $\eta = -a \sin\xi$  is minimum, which is when  $\xi = \pi/2$ , so that at the surface the water

$$\begin{aligned} \frac{\vec{u}}{v_p} &= ka e^{kz} [-1\hat{x} + 0\hat{z}] \\ &= -ka e^{kz} \hat{x} \\ &= -ka e^{k(-a)} \hat{x} \\ &= -ka e^{-ka} \hat{x} \end{aligned}$$

(b) The peak is when  $\eta$  is a maximum so  $\xi = -\pi/2$ .

$$\begin{aligned} \frac{\vec{u}}{v_p} &= ka e^{kz} [-\sin(-\pi/2)\hat{x} + 0\hat{z}] \\ &= ka e^{ka} \hat{x} \end{aligned}$$

(c) We see that the ratio only depends on the quantity  $ka$ , thus the ratio of the amplitude and the wavelength.

4.17 We start by writing out what we know.

$$\eta = -a \sin \xi \quad \longrightarrow \quad \frac{\partial \eta}{\partial t} = a\omega \cos \xi \quad \text{AND} \quad \frac{\partial \eta}{\partial x} = -ka \cos \xi$$

$$\phi = \frac{a\omega}{k} \frac{\cosh \zeta}{\sinh(kd)} \cos \xi$$

$$\frac{\partial \phi}{\partial x} = a\omega \frac{-\cosh \zeta \sin \xi}{\sinh(kd)}$$

$$\frac{\partial \phi}{\partial z} = a\omega \frac{\sinh \zeta \cos \xi \hat{z}}{\sinh(kd)}$$

$$|\nabla \phi|^2 = a^2 \omega^2 \frac{\sinh^2 \zeta + \sin^2 \xi}{\sinh(kd)}$$

By pugging in we find that the BCs become.

$$\frac{\omega^2}{gk} \left[ \frac{\cosh \zeta_\eta}{\sinh(kd)} \sin \xi + \frac{1}{2}ka \frac{\sinh^2 \zeta_\eta + \sin^2 \xi}{\sinh^2(kd)} \right] = \sin \xi \quad \text{Dynamic}$$

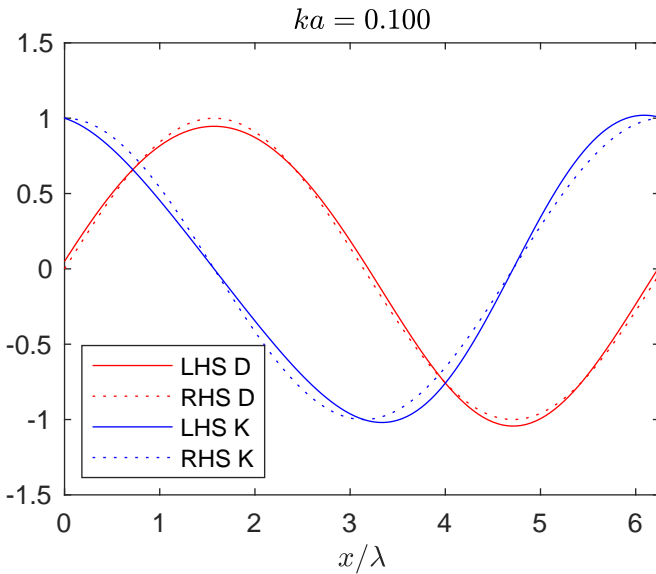
$$\left[ \frac{\sinh \zeta_\eta \cos \xi}{\sinh(kd)} - ka \cos \xi \frac{\cosh \zeta_\eta \sin \xi}{\sinh(kd)} \right] = \cos \xi \quad \text{Kinematic}$$

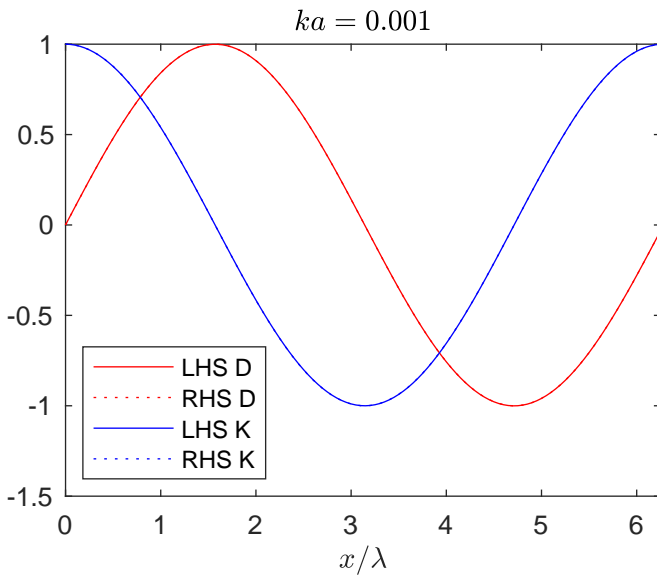
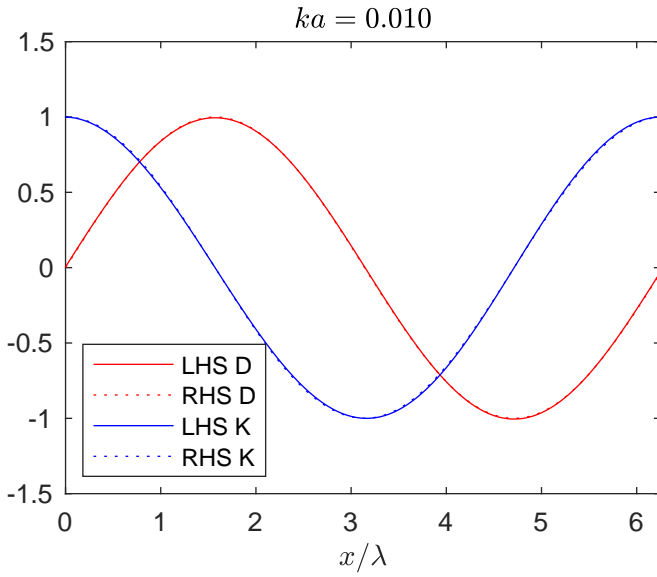
with  $\zeta_\eta = k\eta + kd = -ka \sin \xi + kd$ . Now we also use the restriction that  $\omega^2 = gk \tanh(kd)$  to convert the dynamic BC to the following

$$\left[ \frac{\cosh \zeta_\eta}{\cosh(kd)} \sin \xi + \frac{1}{2}ka \frac{\sinh^2 \zeta_\eta + \sin^2 \xi}{\sinh(kd) \cosh(kd)} \right] = \sin \xi \quad \text{Dynamic}$$

$$\left[ \frac{\sinh \zeta_\eta \cos \xi}{\sinh(kd)} - ka \cos \xi \frac{\cosh \zeta_\eta \sin \xi}{\sinh(kd)} \right] = \cos \xi \quad \text{Kinematic}$$

We notice that the terms we ignored before all have a coefficient  $ka$  while all the other terms are of order 1.





```

[1] function sanityCheck(ka)
[2] kd = pi;
[3] xi = 0:0.05:2*pi;
[4] s = sin(xi);
[5] c = cos(xi);
[6] zeta = -ka*s+kd;
[7] cz = cosh(zeta);
[8] sz = sinh(zeta);
[9] cd = cosh(kd);
[10] sd = sinh(kd);
[11] LHS_D = cz.*s/cd + 0.5*ka*(sz.^2+s.^2)/(sd*cd);
[12] RHS_D = s;
[13] LHS_K = sz.*c/sd - ka*c.*cz.*s/sd;
[14] RHS_K = c;
[15] plot(xi,LHS_D,'r',xi,RHS_D,':r')
[16] hold on

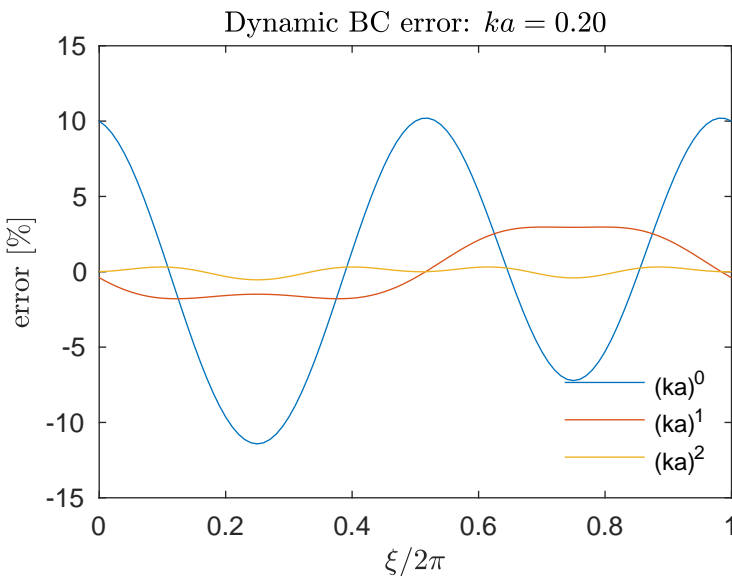
```

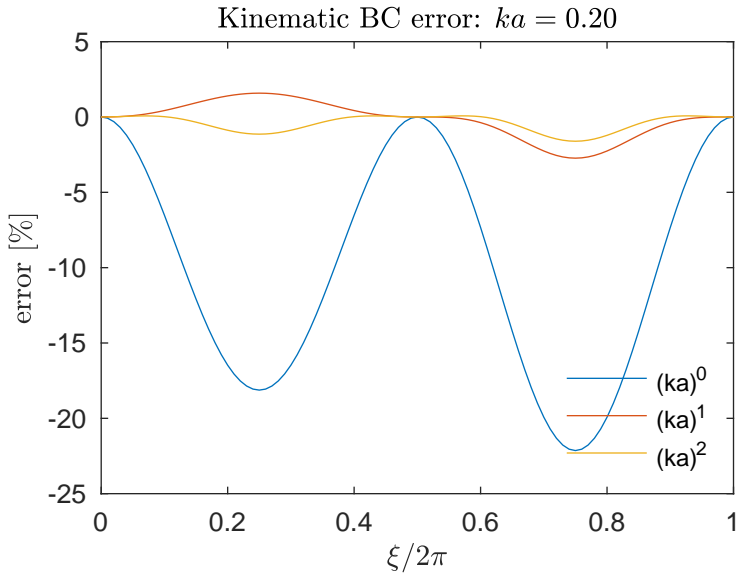
```
[17] plot(xi,LHS_K,'b',xi,RHS_K,':b')
[18] hold off
[19] leg = {'LHS D','RHS D','LHS K','RHS K'};
[20] legend(leg,'location','southwest')
[21] texText('$x/\lambda$', 'x')
[22] texText(sprintf('$ka = %0.3f$',ka), 't')
[23] filename = sprintf('comp%03d',1000*ka);
[24] graphToPDF(filename,4,3)
[25] end
```

5.1 The following code produced the graphs.

```
[1] function graphError2(ka)
[2] xi = 2*pi*(0:0.01:1)';
[3] S = sin(xi);
[4]
[5] eta = NaN(length(xi),3); % really eta/a
[6] eta(:,1) = -S; % zero order
[7] eta(:,2) = -(S+0.5*ka)./(1+ka*S); % first order
[8] eta(:,3) = -S - 0.5*ka*(1-2*S.^2) + 0.5*ka^2*(S-3*S.^3); % second order
[9] ez = exp(ka*eta);
[10]
[11] K = [0, -0.5*ka, -0.5*ka]; %[zero order, first order, second order]
[12] rat = [1, 1, 1/(1-ka^2)];
[13] LHS_D = rat.*ez.*(S + 0.5*ka*ez)+eta;
[14] LHS_K = -K + ez.*S + eta;
[15]
[16] figure(3)
[17] plot(xi/(2*pi),100*LHS_D)
[18] text(sprintf('Dynamic BC error: $ka = %0.2f$',ka),'t')
[19] text('\xi/2\pi','x')
[20] text('error [%]','y')
[21] legend({'(ka)^0','(ka)^1','(ka)^2'},'location','southeast','box','off')
[22] graphToPDF(sprintf('errorD%0.2f.pdf',ka),4.25,3)
[23]
[24] figure(4)
[25] plot(xi/(2*pi),100*LHS_K)
[26] text(sprintf('Kinematic BC error: $ka = %0.2f$',ka),'t')
[27] text('\xi/2\pi','x')
[28] text('error [%]','y')
[29] legend({'(ka)^0','(ka)^1','(ka)^2'},'location','southeast','box','off')
[30] graphToPDF(sprintf('errorK%0.2f.pdf',ka),4.25,3)
[31] end
```

(a)





(b) The problem is when  $\sin \xi \rightarrow 0$  because in this case the denominator becomes zero. The limit of the ratio is still finite since the numerator is also zero, but this may become unreliable since the numerator is the difference of two finite things that get closer and closer as  $\sin \xi \rightarrow 0$ . Eventually the round-off error of the calculation will kill the accuracy of the difference, making the results unreliable.



**6.1** In general to compute the gravitational potential energy of a distributed object we take the volume integral over the extent of the object

$$U = \int dm g z = \int \rho dV g z = \rho g \int dV z$$

where  $z$  is the vertical coordinate of the volume element  $dV$ . Thus if we take the wave to be described by  $\eta(x) = -a \sin(kx)$  then the negative part is the first half wavelength of  $x$ . Let us consider the section that is colored dark blue before, and extending a distance  $w$  in the  $y$  direction (into the paper), the volume element will be  $dV = dx dy yz$  and the initial potential energy will be

$$\begin{aligned} U_i &= \rho g \int_0^w dy \int_0^{L/2} dx \int_{\eta(x)}^0 dz z \\ &= \rho g \int_0^w dy \int_0^{L/2} dx \frac{1}{2} [z^2]_{\eta(x)}^0 \\ &= -\frac{\rho g}{2} \int_0^w dy \int_0^{L/2} dx \eta^2(x) \\ &= -\frac{\rho g}{2} w \int_0^{L/2} dx \eta^2(x) \end{aligned}$$

while the final potential energy will be

$$\begin{aligned} U_f &= \rho g \int_0^w dy \int_{L/2}^L dx \int_0^{\eta(x)} dz z \\ &= \rho g \int_0^w dy \int_{L/2}^L dx \frac{1}{2} [z^2]_0^{\eta(x)} \\ &= \frac{\rho g}{2} \int_0^w dy \int_{L/2}^L dx \eta^2(x) \\ &= \frac{\rho g}{2} w \int_{L/2}^L dx \eta^2(x) \end{aligned}$$

The potential energy of the light blue part does not change so the total change in potential energy of the water is

$$\begin{aligned} U &= U_f - U_i = \frac{\rho g}{2} w \int_{L/2}^L dx \eta^2(x) - \left(-\frac{\rho g}{2} w\right) \int_0^{L/2} dx \eta^2(x) \\ &= \frac{\rho g}{2} w \int_{L/2}^L dx \eta^2(x) + \frac{\rho g}{2} w \int_0^{L/2} dx \eta^2(x) \\ &= \frac{\rho g}{2} w \int_0^L dx \eta^2(x) \end{aligned}$$

Thus the potential energy per surface area is

$$\mathbf{U} = \frac{U}{A} = \frac{U}{wL} = \frac{\rho g}{2} \frac{1}{L} \int_0^L dx \eta^2(x) = \frac{\rho g}{2} \langle \eta^2 \rangle$$

The actual shape of  $\eta$  does not matter really as long the volume of the before and after sections are the same and this is assured since  $z = 0$  is defined to be the location of the mean sea level. For the particular case of  $\eta = -a \sin(x)$  we know that  $\langle \eta \rangle = \langle a^2 \sin^2(x) \rangle = \frac{1}{2} a^2$ , and so

$$\mathbf{U} = \frac{\rho g a^2}{4}$$

## 6.2

(a) We start with the result that

$$\langle \mathcal{P} \rangle_t = \mathbf{E} \omega \int dA e^{2kz}$$

So that the power through a strip  $w$  wide and going from the surface to a depth  $\ell$ . Then we can write

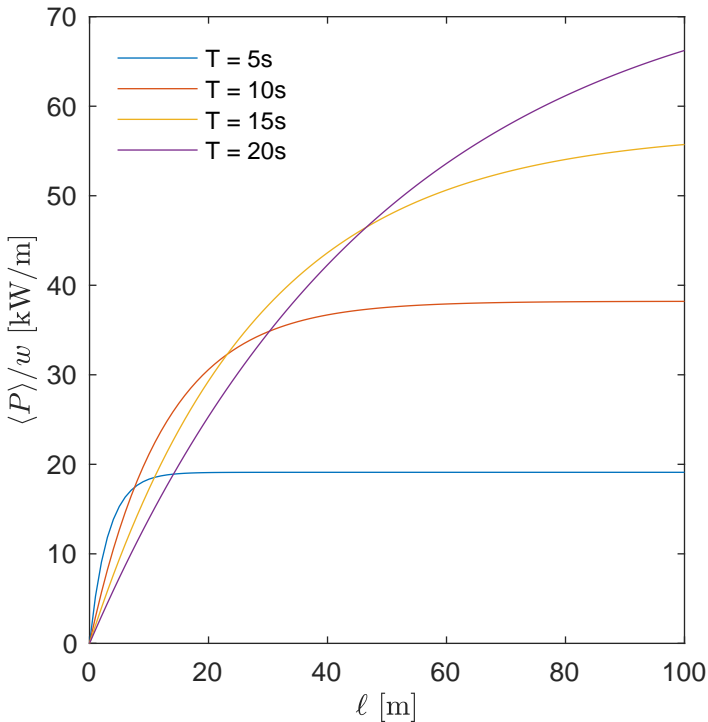
$$\begin{aligned} \langle \mathcal{P} \rangle_t &= \mathbf{E} \omega \int dA e^{2kz} \\ &= \mathbf{E} \omega \int_0^w dy \int_{-\ell}^0 dz e^{2kz} \\ &= \mathbf{E} \omega w \left[ \frac{e^{2kz}}{2k} \right]_{-\ell}^0 \\ &= \mathbf{E} \omega w \frac{1 - e^{-2k\ell}}{2k} \end{aligned}$$

and so

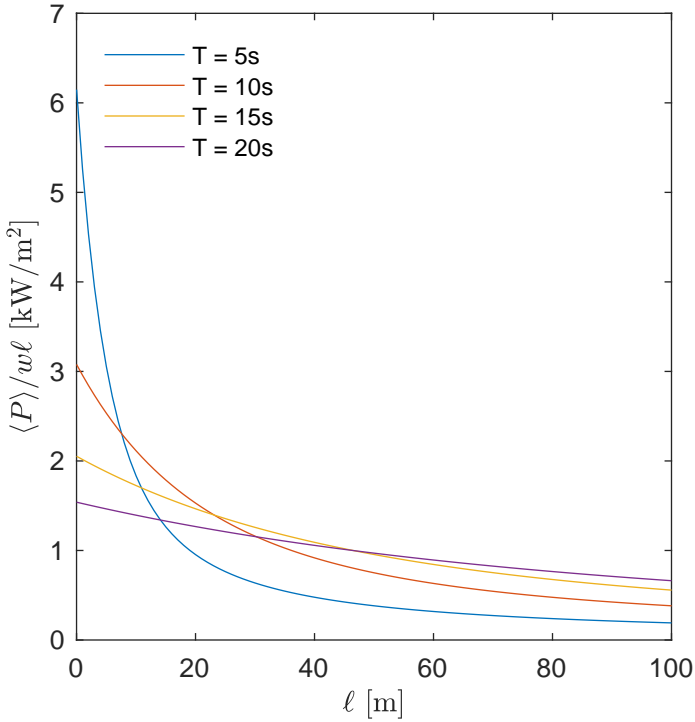
$$\frac{\langle \mathcal{P} \rangle_t}{w\ell} = \mathbf{E} \omega \frac{1 - e^{-2k\ell}}{2k\ell}$$

(b) Since  $\mathbf{E} = \frac{\rho g a^2}{2}$  we see that the power increases as the square of the amplitude. Setting  $a$  to one meter we find  $\mathbf{E} = \frac{\rho g a^2}{2} = 4.9 \frac{\text{kJ}}{\text{m}^2}$

(c)



(d)



(e) From the graph of power per area we see that the power per area decreases with  $\ell$  thus it is better to have a wide and shallow panel.

(f) With a  $\ell$  small we can use the expansion  $e^{-2k\ell} \approx 1 - 2k\ell$  and thus

$$\begin{aligned} \frac{\langle P \rangle}{w\ell} &= E \omega \frac{1 - e^{-2k\ell}}{2k\ell} \\ &= E \omega \frac{1 - (1 - 2k\ell)}{2k\ell} \\ &= E \omega = E \frac{2\pi}{T} > 385 \frac{\text{W}}{\text{m}^2} \end{aligned}$$

where we have used the lowest energy period of 20 seconds, and we have reduced the value of  $E$  to  $E = 0.5^2(4.9 \frac{\text{kJ}}{\text{m}^2})$  since our amplitude  $a$  is half a meter instead of one meter. Thus with an area of 1 square meter and a minimum power generate of 80 watts we have

$$\alpha = \frac{80\text{W}}{385\text{W}} = 0.21$$

a five percent efficiency.

(g) The maximum incident power of 1000 watts leads to a maximum average power to  $\frac{1000}{\pi}$  watts. With an efficiency of 37% this will generate a power of 118 watts. This is a bit better than the 80 watts produced by our 21 % efficient wave energy sucker.

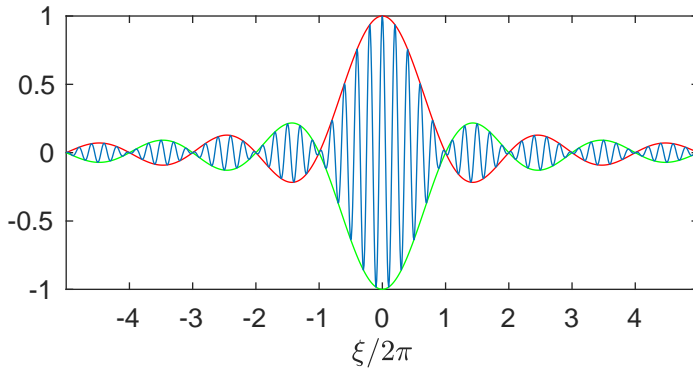
6.3

6.4 In this case we have

$$\int a(\kappa)e^{i\kappa\xi}d\kappa = \int_{-1/2}^{1/2} 1e^{i\kappa\xi}d\kappa = \text{sinc}(\xi/2)$$

and so

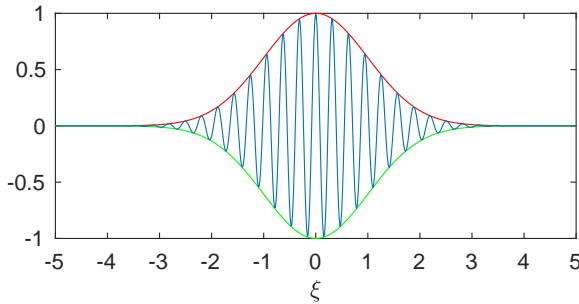
$$\eta = \cos(\bar{k}x - \bar{\omega}t)\text{sinc}(\xi/2)$$



6.5 We first compute

$$\begin{aligned}
 \psi &= \int_{-\infty}^{\infty} a(k) e^{i(kx - \omega t)} dk \\
 &= e^{i(\bar{k}x - \bar{\omega}t)} \int_{-\infty}^{\infty} a(k) e^{i((k - \bar{k})x - (\omega - \bar{\omega})t)} dk \\
 &= e^{i(\bar{k}x - \bar{\omega}t)} \int_{-\infty}^{\infty} a(k) e^{i((k - \bar{k})x - (k - \bar{k})v_g t)} dk \\
 &= e^{i(\bar{k}x - \bar{\omega}t)} \int_{-\infty}^{\infty} a(k) e^{i(k - \bar{k})(x - v_g t)} dk \\
 &= e^{i(\bar{k}x - \bar{\omega}t)} \int_{-\infty}^{\infty} a(\kappa) e^{i\kappa\xi} \sigma_k d\kappa \\
 &= e^{i(\bar{k}x - \bar{\omega}t)} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\kappa^2} e^{i\kappa\xi} d\kappa \\
 &= e^{i(\bar{k}x - \bar{\omega}t)} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\kappa^2 + i\kappa\xi} d\kappa \\
 &= e^{i(\bar{k}x - \bar{\omega}t)} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\kappa^2 - 2i\kappa\xi)} d\kappa \\
 &= e^{i(\bar{k}x - \bar{\omega}t)} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\kappa^2 - 2i\kappa\xi + (i\xi)^2)} e^{\frac{1}{2}(i\xi)^2} d\kappa \\
 &= e^{i(\bar{k}x - \bar{\omega}t)} e^{\frac{1}{2}(i\xi)^2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\kappa - i\xi)^2} d\kappa \\
 &= e^{i(\bar{k}x - \bar{\omega}t)} e^{-\frac{1}{2}\xi^2}
 \end{aligned}$$

with  $\kappa = \frac{k - \bar{k}}{\sigma_k}$  and  $\xi = \sigma_k(x - v_g t)$ . Thus  $\eta = \cos(\bar{k}x - \bar{\omega}t) e^{-\frac{1}{2}\xi^2}$  and we see that the envelope is a gaussian in  $\xi$ !



```

[1] function gauss(sigma_k,kbar)
[2] lam = 2*pi/kbar;
[3] dx = lam/20;
[4] sigma_x = 1/sigma_k;
[5] x = -5*sigma_x:dx:5*sigma_x;
[6] xi = sigma_k*x;
[7] env = exp(-0.5*xi.^2);
[8] eta = cos(kbar*x).*env; % wave at t=0
[9] plot(xi,env,'r',xi,-env,'g',xi,eta)
[10] text(xi,'xi','x')
[11] xticks(-5:5)
[12] graphToPDF('gauss.pdf',4.25,2)
[13] end

```

6.7 The following code makes the graphs.

```
[1] function [hist_H,hist_x] = buoyHist(day)
[2]     switch day
[3]         case 'march2'
[4]             fn = '../..../waveData/xy029p101_201903020000-201903032359'; %Day of
visit
[5]         case 'march1'
[6]             fn = '../..../waveData/xy029p101_201903010000-201903022359'; %Day of
fore
[7]         otherwise
[8]             disp('hey dummy that is not an option. I dont know how to spell b
you dont know how to type.');
```

```
[9]             return;
[10]        end
[11]        [r,stepError] = readBuoyFile(fn);
[12]        fs = 1.28;
[13]        N30 = 30*60*fs;
[14]        Ns = size(r,1);
[15]        Ng = floor(Ns/N30);
[16]        % Ng = 5;
[17]        zg = reshape(r(1:Ng*N30,3)/100,N30,Ng);
[18]        error = sum(reshape(stepError(1:Ng*N30),N30,Ng));
[19]        hist_H = NaN(14,Ng);
[20]        Hm      = NaN(Ng,4);
[21]        hist_T = NaN(12,Ng);
[22]        Tm      = NaN(Ng,4);
[23]
[24]        for ng = 1:Ng
[25]            if error(ng)==0
[26]                [hist_H(:,ng),hist_Hx,hist_T(:,ng),hist_Tx,Hm(ng,:),Tm(ng,:)] = c
[27]            end
[28]        end
[29]        figure(1)
[30]        imagesc([1,Ng],hist_Hx,hist_H)
[31]        hold on
[32]        plot(1:Ng,Hm,'r')
[33]        hold off
[34]        texText('group','x')
[35]        texText('$H$ [m]','y')
[36]        graphToPDF('hist_height_all.pdf',4,4)
[37]
[38]        figure(10)
[39]        imagesc([1,Ng],hist_Tx,hist_T)
[40]        hold on
[41]        plot(1:Ng,Tm,'r')
[42]        hold off
[43]        texText('group','x')
[44]        texText('$T$ [s]','y')
[45]        graphToPDF('hist_period_all.pdf',4,4)
[46]    end
[47]
[48]
[49] function [hist_H,hist_Hx,hist_T,hist_Tx,Hm,Tm] = comp(z,ng)
[50]     fs = 1.28;
```

```

[51] dt = 1/fs;
[52] Ns = size(z,1);
[53] NcUB = round(Ns/5);
[54] ind = NaN(1,NcUB);
[55] tc = NaN(1,NcUB);
[56] Nc = 0;
[57] for n=1:Ns-1
[58]     if z(n)>=0 && z(n+1)<0
[59]         Nc = Nc+1;
[60]         ind(Nc) = n;
[61]         tc(Nc) = (n+z(n)/(z(n)-z(n+1)))*dt;
[62]     end
[63] end
[64] ind = ind(1:Nc);
[65] tc = tc(1:Nc);
[66]
[67] T = NaN(1,Nc-1);
[68] H = NaN(1,Nc-1);
[69]
[70] for n = 1:Nc-1
[71]     rng = (ind(n)+1):ind(n+1);
[72]     T(n) = tc(n+1)-tc(n);
[73]     z_section = z(rng);
[74]     H(n) = max(z_section) - min(z_section);
[75] end
[76]
[77] ind = find(isfinite(H));
[78] H = H(ind);    %remove the NaN elements of H and T
[79] T = T(ind);
[80] [~,ind] = sort(H,'descend');
[81] H = H(ind);    % now put in descending order
[82] T = T(ind);
[83]
[84] Nf = length(H);
[85] N_third = round(Nf/3);
[86] N_tenth = round(Nf/10);
[87]
[88] H_mean = mean(H);
[89] H_rms = sqrt(mean(H.^2));
[90] H_third = mean(H(1:N_third));
[91] H_tenth = mean(H(1:N_tenth));
[92]
[93] T_mean = mean(T);
[94] T_rms = sqrt(mean(T.^2));
[95] T_third = mean(T(1:N_third));
[96] T_tenth = mean(T(1:N_tenth));
[97]
[98]
[99] figure(2)
[100] subplot(2,1,1)
[101]     dH = 0.3;
[102]     h = histogram(H,dH/2:dH:4.5,'displaystyle','stairs');
[103]     hold on
[104]     A = 1;

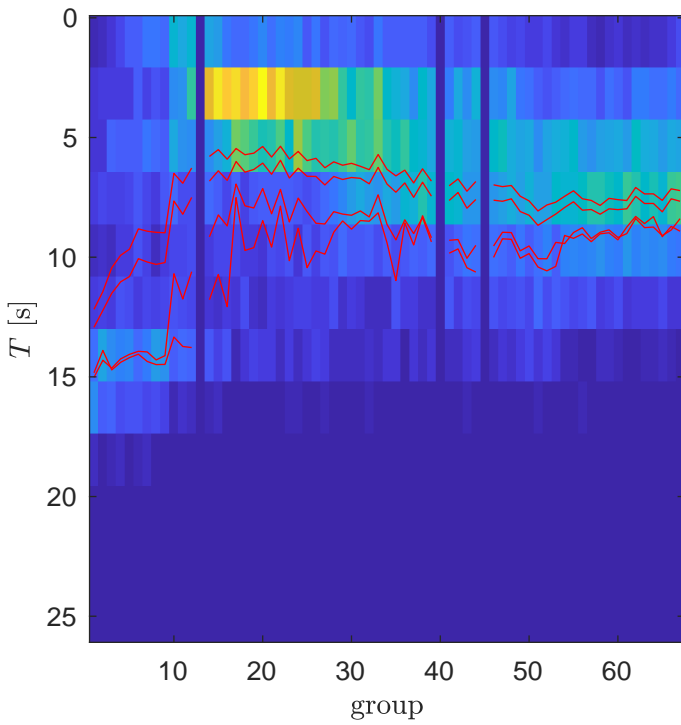
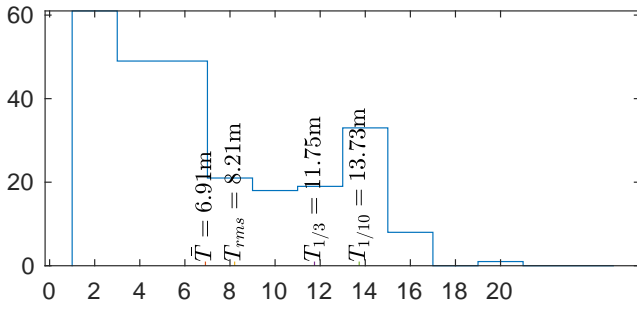
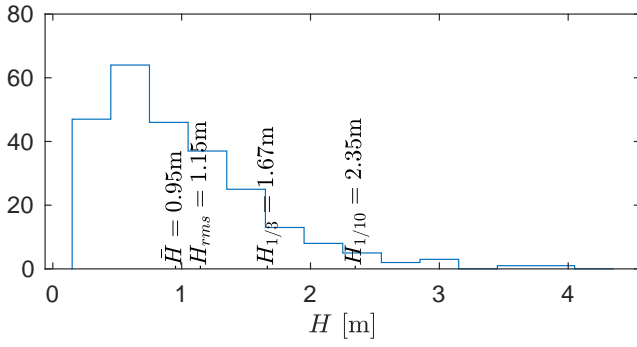
```

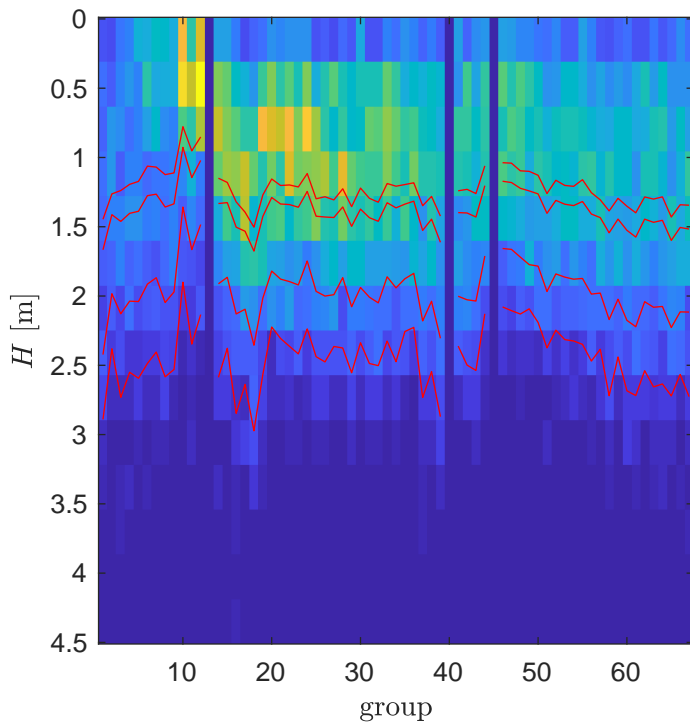


```

[105]     plot(H_mean * [1,1], [0,A], 'k');
[106]     plot(H_rms   * [1,1], [0,A], 'k');
[107]     plot(H_third * [1,1], [0,A], 'k');
[108]     plot(H_tenth * [1,1], [0,A], 'k');
[109]     hold off
[110]     text(H_mean ,A,sprintf( '$\bar H=%0.2f\$m',H_mean ),'rotation',90,'in
[111]     text(H_rms   ,A,sprintf('$H_{rms }=%0.2f\$m',H_rms   ),'rotation',90,'in
[112]     text(H_third ,A,sprintf('$H_{1/3 }=%0.2f\$m',H_third),'rotation',90,'in
[113]     text(H_tenth ,A,sprintf('$H_{1/10}=%0.2f\$m',H_tenth),'rotation',90,'in
[114]     xticks(0:1:5)
[115]     texText('$H$ [m]', 'x')
[116]     hist_H = h.Values;
[117]     hist_Hx = h.BinEdges;
[118]     Hm = [H_mean,H_rms,H_third,H_tenth];
[119]
[120] subplot(2,1,2)
[121]     dT = 2;
[122]     h = histogram(T,dT/2:dT:25,'displaystyle','stairs');
[123]     hold on
[124]     A = 1;
[125]     plot(T_mean * [1,1], [0,A]);
[126]     plot(T_rms   * [1,1], [0,A]);
[127]     plot(T_third * [1,1], [0,A]);
[128]     plot(T_tenth * [1,1], [0,A]);
[129]     hold off
[130]     text(T_mean ,A,sprintf( '$\bar T=%0.2f\$m',T_mean ),'rotation',90,'in
[131]     text(T_rms   ,A,sprintf('$T_{rms }=%0.2f\$m',T_rms   ),'rotation',90,'in
[132]     text(T_third ,A,sprintf('$T_{1/3 }=%0.2f\$m',T_third),'rotation',90,'in
[133]     text(T_tenth ,A,sprintf('$T_{1/10}=%0.2f\$m',T_tenth),'rotation',90,'in
[134]     xticks(0:2:20)
[135]     hist_T = h.Values;
[136]     hist_Tx = h.BinEdges;
[137]     Tm = [T_mean,T_rms,T_third,T_tenth];
[138]
[139] graphToPDF(sprintf('histHT%d.pdf',ng),4.5,4.5)
[140] end
[141]

```

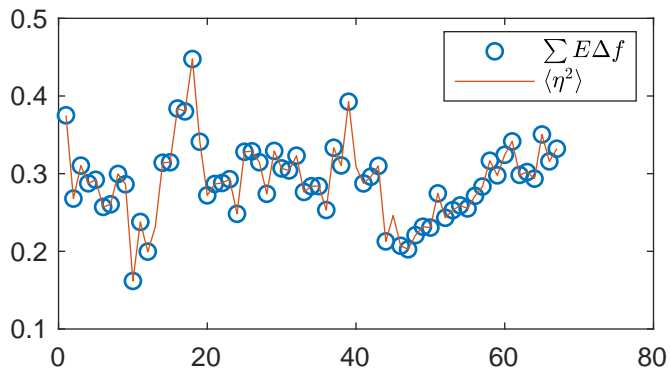




6.8 The following code will make the graphs.

```
[1] function [sum_Edf,dist_E] = buoyDist(day)
[2]     switch day
[3]         case 'march2'
[4]             fn = '../..../waveData/xy029p101_201903020000-201903032359'; %Day o
visit
[5]         case 'march1'
[6]             fn = '../..../waveData/xy029p101_201903010000-201903022359'; %Day b
fore
[7]         otherwise
[8]             disp('hey dummy that is not an option. I dont know how to spell b
you dont know how to type.');
```

```
[9]             return;
[10]     end
[11]     [r,stepError] = readBuoyFile(fn);
[12]     fs = 1.28;
[13]     N30 = 30*60*fs;
[14]     Ns = size(r,1); % number of smaples.
[15]     Ng = floor(Ns/N30); % number of 30 minute groups of samples.
[16]     Nfft = 128; % the rest assumes N30/Nfft is an integer.
[17]     Ne = N30/Nfft; % number of FFT's in a 30 minute ensamble.
[18]     eta = reshape(r(1:Ng*N30,3)/100,Nfft,Ne,Ng);
[19]     eta = eta - mean(eta,1);
[20]     error = sum(reshape(stepError(1:Ng*N30),N30,Ng));
[21]
[22]     mean_eta2 = squeeze(mean(mean(eta.^2)));
[23]
[24]     dist_E = NaN(Nfft/2,Ng);
[25]
[26]     df = fs/Nfft;
[27]     for ng = 1:Ng
[28]         if error(ng)==0
[29]             H = fft(squeeze(eta(:,:,ng)));
[30]             dist_E(:,ng) = (2/(Nfft*fs))*mean(abs(H((1:Nfft/2)+1,:)).^2,2);
[31]         end
[32]     end
[33]     sum_Edf = sum(dist_E)*df;
[34]
[35]     figure(1)
[36]     surf(10*log10(dist_E))
[37]
[38]     figure(3)
[39]     ng = 1:Ng;
[40]     plot(ng,sum_Edf, 'o',ng, mean_eta2)
[41]     legend({'$\sum E\Delta f$', '$\langle \eta^2 \rangle$'}, 'interpreter', 'latex'
[42]     graphToPDF('varDensity.pdf',4,2)
[43] end
```



6.9 Using the given  $p(a_n)$

$$p(a) = \frac{1}{\sigma^2} a e^{-a^2/2\sigma^2}$$

Thus

$$\begin{aligned}\mu = \langle a \rangle &= \int_0^\infty a p(a) da \\ &= \int_0^\infty \frac{a^2}{\sigma^2} e^{-a^2/2\sigma^2} da \\ &= \sigma \int_0^\infty \frac{a^2}{\sigma^2} e^{-a^2/2\sigma^2} \frac{da}{\sigma} \\ &= \sigma \int_0^\infty u^2 e^{-u^2/2} du \\ &= \sigma \sqrt{\frac{\pi}{2}}\end{aligned}$$

so  $\langle a_n \rangle_e = \sqrt{\frac{\pi}{2}} \sigma_n$ .

6.10 Using the given  $p(\varepsilon_n)$ .

$$\begin{aligned}
 E_n &= E(f_n) = \langle \varepsilon_n \rangle \\
 &= \int_0^\infty \varepsilon_n p(\varepsilon_n) d\varepsilon_n \\
 &= \int_0^\infty \varepsilon_n \frac{\Delta f}{\sigma_n^2} e^{-\Delta f \varepsilon_n / \sigma_n^2} d\varepsilon_n \\
 &= \frac{\sigma_n^2}{\Delta f} \int_0^\infty \frac{\Delta f \varepsilon_n}{\sigma_n^2} e^{-\Delta f \varepsilon_n / \sigma_n^2} \frac{\Delta f}{\sigma_n^2} d\varepsilon_n \\
 &= \frac{\sigma_n^2}{\Delta f} \int_0^\infty u e^{-u} du \\
 &= \frac{\sigma_n^2}{\Delta f}
 \end{aligned}$$

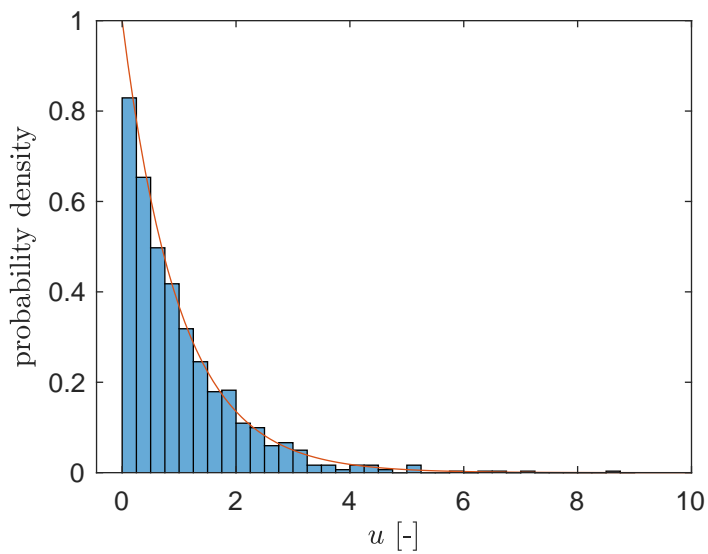
and so

$$p(\varepsilon_n) = \frac{1}{E_n} e^{-\varepsilon_n / E_n}$$

6.11 This code creates the graph

```
[1] function buoyDistFine()
[2]     fn = '.././waveData/xy029p101_201903020000-201903032359';
[3]     [r,stepError] = readBuoyFile(fn);
[4]     fs = 1.28;
[5]     N30 = 60*30*fs;
[6]     Ns = size(r,1);      % number of samples.
[7]     Ng = floor(Ns/N30); % number of 30 minute groups of samples.
[8]     Nfft = 128;         % the rest assumes N30/Nfft is an integer.
[9]     Ne = N30/Nfft;      % number of FFT's in a 30 minute ensemble.
[10]    eta = reshape(r(1:Ng*N30,3)/100,Nfft,Ne,Ng);
[11]    eta = eta - mean(eta,1); % remove mean from each column.
[12]    error = sum(reshape(stepError(1:Ng*N30),N30,Ng));
[13]    dist_E = NaN(Ne,Ng);
[14]    n_p = 8; % The index of the peak amplitude in the fft, f_8 = 0.07 Hz;
[15]    for ng = 1:Ng
[16]        if error(ng)==0
[17]            H = fft(squeeze(eta(:,:,ng)));
[18]            H_peak = H(n_p,:);
[19]            dist_E(:,ng) = (2/(Nfft*fs))*abs(H_peak).^2;
[20]        end
[21]    end
[22]    mu = mean(dist_E);
[23]    u = reshape(dist_E./mu,1,[]);
[24]    u_bin_edges = 0:0.25:9;
[25]    histogram(u,u_bin_edges,'Normalization','pdf')
[26]    hold on
[27]    x = 0:0.1:10;
[28]    plot(x,exp(-x));
[29]    textText('$u$ [-]', 'x')
[30]    textText('probability density', 'y')
[31]    hold off
[32]    graphToPDF('distFine.pdf',4,3)
[33]
[34] end
```





## 6.12

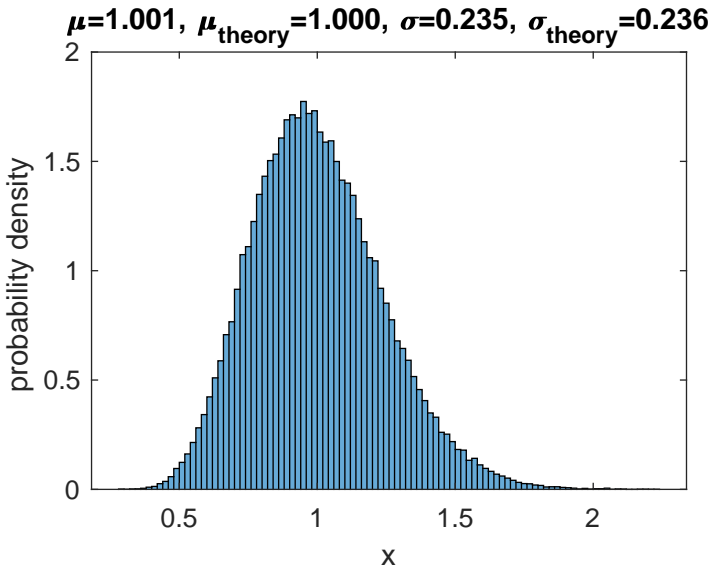
$$\begin{aligned} [x^2] &= \int_0^{\infty} x^2 p(x) dx \\ &= \int_0^{\infty} x^2 \frac{1}{\mu} e^{-x/\mu} dx \\ &= \mu^2 \int_0^{\infty} \frac{x^2}{\mu^2} e^{-x/\mu} \frac{dx}{\mu} \\ &= \mu^2 \int_0^{\infty} u^2 e^{-u} du \\ &= \mu^2 \cdot 2 \end{aligned}$$

$$\sigma_x = \sqrt{[x^2] - [x]^2} = \sqrt{2\mu^2 - \mu^2} = \mu$$

6.13

**6.14** The following code produces the graph and computes the statistics. The graph was produced with  $a = 1$ ,  $N = 18$  and  $M = 100000$ .

```
[1] function statsOfmean(N,M,a)
[2]     x = -a*log(rand(N,M));
[3]     mu = mean(x,1);
[4]     figure(1)
[5]     histogram(mu,'normalization','pdf');
[6]     mean_mu = mean(mu);
[7]     std_mu = std(mu);
[8]     mean_theory = a;
[9]     std_theory = a/sqrt(N);
[10]    str = sprintf('\mu=%0.3f, \mu_{theory}=%0.3f, \sigma=%0.3f, \sigma_{t}
[11]    title(str)
[12]    xlabel('x')
[13]    ylabel('probability density')
[14]    if N==18
[15]    graphToPDF(sprintf('statsOfmean%d.pdf',N),4,3)
[16]    end
[17]
[18] end
```



The mean and standard deviations are printed on the graph, and we see that they are the same for the theory and numerical methods.

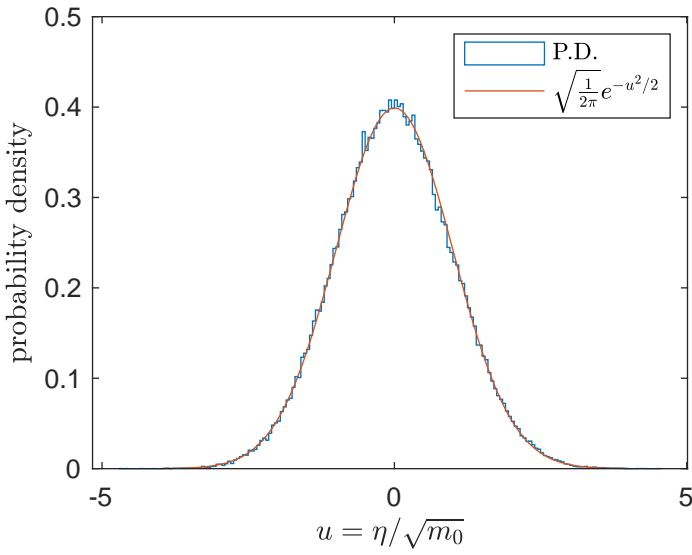
7.1 The following code is a method of doing the analysis, and it produces the following graphs, in addition two other graphs.

```
[1] function buoyStats()
[2] fname = '.././waveData/xy029p101_201903020000-201903032359';
[3] [r,stepError] = readBuoyFile(fname);
[4] fs = 1.28;
[5] N30 = 60*30*fs;      % number of samples in 30 minutes
[6] Ns = size(r,1);     % number of samples.
[7] Ng = floor(Ns/N30); % number of 30 minute groups of samples.
[8] Nfft = 128;        % the rest assumes N30/Nfft is an integer.
[9] Ne = N30/Nfft;     % number of FFT's in a 30 minute ensemble.
[10] eta = reshape(r(1:Ng*N30,3)/100,Nfft,Ne,Ng);
[11] error = sum(reshape(stepError(1:Ng*N30),N30,Ng));
[12] eta = eta(:,:,error==0); % keep just the error free groups.
[13] Ng = size(eta,3);
[14] eta = eta - mean(eta,1); % remove mean from each column.
[15] En = NaN(Nfft/2,Ng);
[16] df = fs/Nfft;
[17] for ng = 1:Ng % compute spectral density and m0 and m2.
[18]     H = fft(squeeze(eta(:,:,ng)),1); % compute FFT of all sections in group
[19]     H = H(1:Nfft/2,:); % cut off the redundant half of the spectrum
[20]     epsilon = (2/(Nfft*fs))*abs(H).^2;
[21]     En(:,ng) = mean(epsilon,2);
[22] end
[23] fn = (0:(Nfft/2-1))*df;
[24] m0 = sum(En,1)*df;
[25] m2 = sum(fn.^2.*En,1)*df;
[26] eta = reshape(eta,Nfft*Ne,Ng); % regroup by half hour rather than Nfft
[27]
[28] % Compute the mean period for each value of eta threshold
[29] eta_lim = 0:0.01:1.6;
[30] meanT = NaN(length(eta_lim),Ng);
[31] for ng = 1:Ng
[32]     if error(ng)==0
[33]         meanT(:,ng) = meanPeriod(eta(:,ng),eta_lim,fs);
[34]     end
[35] end
[36] upsilon = m0.*log(meanT.^2.*(m2./m0));
[37]
[38] figure(1) % graph the pdf of eta
[39] histogram(eta./sqrt(m0),'normalization','pdf','displaystyle','stairs')
[40] hold on
[41] u = -4:0.01:4;
[42] plot(u,sqrt(1/(2*pi))*exp(-0.5*u.^2))
[43] hold off
[44] texText('$u = \eta/\sqrt{m_0}$','x')
[45] texText('probability density','y')
[46] legend({'P.D.', '$\sqrt{1\over 2\pi}e^{-u^2/2}$'},'interpreter','latex')
[47] graphToPDF('etaPDF.pdf',4,3)
[48]
[49] figure(2) % plot Upsilon verus eta
[50] plot(eta_lim,upsilon)
[51] hold on
[52] plot(eta_lim,eta_lim.^2,'g:','linewidth',2)
```

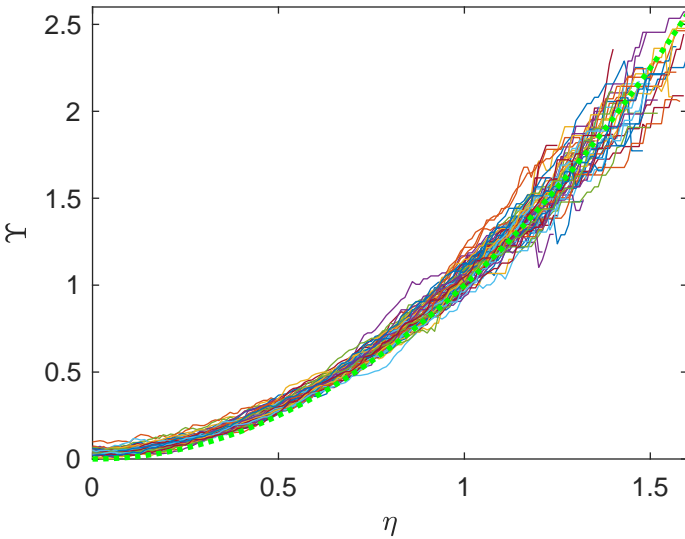
```

[53]     hold off
[54]     texText('$\eta$', 'x')
[55]     texText('$\Upsilon$', 'y')
[56]     xlim([0,1.6])
[57]     ylim([0,2.6])
[58]     graphToPDF('LHSvsEta.pdf',4,3)
[59]
[60] figure(3) % a 2-D histogram of (eta,upsilon)
[61]     eta_m = eta_lim'*ones(1,size(upsilon,2));
[62]     eta_m = reshape(eta_m, [],1);
[63]     upsilon    = reshape(upsilon    , [],1);
[64]     xEdges = eta_lim-0.001;
[65]     yEdges = 0:0.01:2.5;
[66]     histogram2(eta_m,upsilon,xEdges,yEdges,'displaystyle','tile')
[67]     hold on
[68]     plot(eta_lim,eta_lim.^2,'r:','linewidth',1)
[69]     hold off
[70]     texText('$\eta$', 'x')
[71]     texText('$\Upsilon$', 'y')
[72]     texText('Histogram','t')
[73]     graphToPDF('LHS-PDF.pdf',4,4);
[74]
[75] figure(4) % a 2-D histogram of (eta,sqrt(upsilon))
[76]     yEdges = 0:0.005:1.6;
[77]     histogram2(eta_m,sqrt(upsilon),xEdges,yEdges,'displaystyle','tile')
[78]     hold on
[79]     plot(eta_lim,eta_lim,':','linewidth',2)
[80]     hold off
[81]     texText('$\eta$', 'x')
[82]     texText('$\sqrt{\Upsilon}$', 'y')
[83]     texText('Histogram','t')
[84]
[85] figure(5) % a surface plot of upsilon verus eta over time
[86]     surf(upsilon)
[87]
[88] end
[89]
[90] function T = meanPeriod(eta,eta_lim,fs)
[91] % computes the mean period of eta for each threshold value in the
[92] % vector eta_lim
[93]     N = length(eta_lim);
[94]     T = NaN(1,N);
[95]     for n = 1:N
[96]         ind = find(diff(eta>eta_lim(n))==(-1)); % find level crossings
[97]         NT = length(ind)-1; % the number of periods found
[98]         if NT>6 % if there is enough periods for trusty mean
[99]             T(n) = (1/fs)*(ind(end)-ind(1))/NT;
[100]         end
[101]     end
[102] end

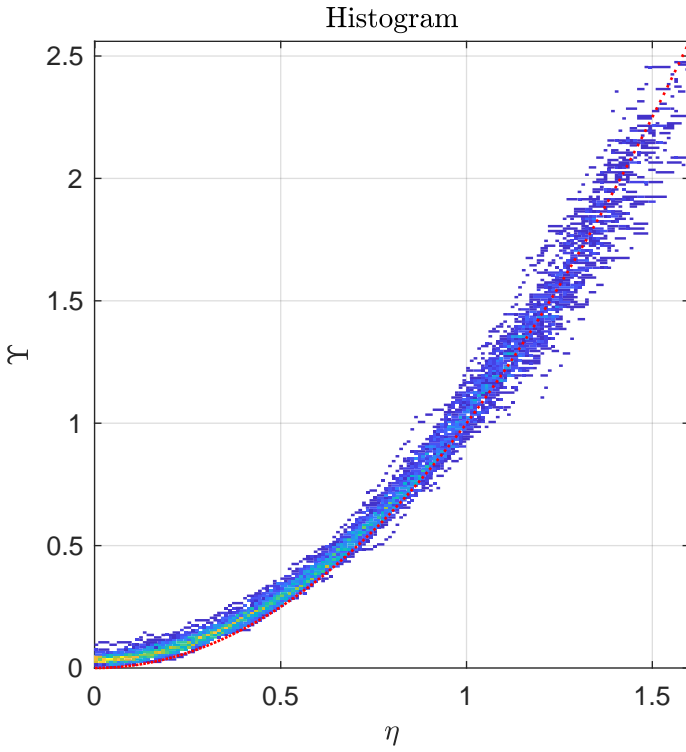
```



In the above graph we see the probability density of  $u = \eta/\sqrt{m_0}$  and the gaussian  $\sqrt{\frac{1}{2\pi}}e^{-u^2/2}$ . We can see that the probability density follows closely the gaussian.



In the above graph we see  $\Upsilon$  versus  $\eta$ . There is one graph for each half hour of data. The green dotted line is  $y = \eta^2$  for reference. We see that while there is some systematic difference from the theoretical relationship  $\Upsilon = \eta^2$  that the trend is very close to the theoretical.

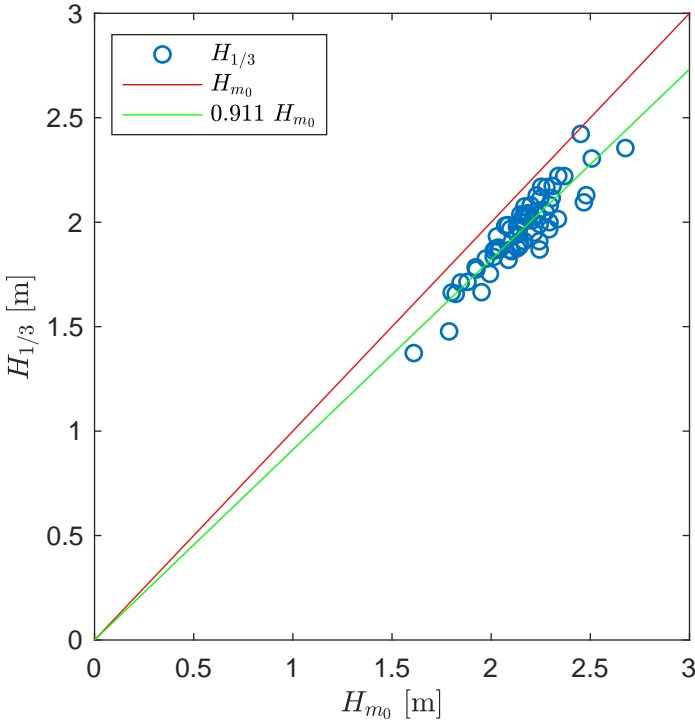


The above graph is a histogram of the same data as in the previous graph. Because there were so many graphs on top of each other it was hard to see the density of lines in the previous graph. This graph shows in some sense the density of lines, so it is easier to see the general trend. The theoretical curve is the red dotted line. The deviation from this curve is greatest near  $\eta = 0$ , but again the overall trend is very close to the theoretical.



7.2 The following code produces the desired graph.

```
[1] function buoySWH()
[2] fname = '.././waveData/xy029p101_201903020000-201903032359';
[3] [r,stepError] = readBuoyFile(fname); % read data file
[4] fs = 1.28; % the sample rate
[5] N30 = 60*30*fs; % number of samples in 30 minutes
[6] Ns = size(r,1); % number of samples.
[7] Ng = floor(Ns/N30); % number of 30 minute groups of samples.
[8] eta = r(1:Ng*N30,3); % Truncate data to full group.
[9] eta = eta/100; % Convert data from cm to meters.
[10] eta = reshape(eta,N30,Ng); % Group data.
[11] error = stepError(1:Ng*N30); % Truncate error data to full group.
[12] error = sum(reshape(error,N30,Ng)); % Group error.
[13] eta = eta(:,error==0); % Keep just the error free groups.
[14] eta = eta - mean(eta,1); % remove mean from each group.
[15] m0 = mean(eta.^2,1); % Compute the mean squared amplitude.
[16] H_m0 = 4*sqrt(m0); % Compute H_m0.
[17]
[18] Ng = length(m0); % Get number of groups.
[19] H_third = NaN(1,Ng); % Allocate an array to store H_1/3.
[20] for ng = 1:Ng % Loop over groups.
[21] H_third(ng) = comp_H_third(eta(:,ng)); % Compute H_1/3.
[22] end
[23] slope = sum(H_m0.*H_third)/sum(H_m0.^2); % Compute least squares slope.
[24] figure(1) % Graph H_1/3 versus H_m0.
[25] x = [0,3]; % Set range of H_m0 to graph.
[26] plot(H_m0,H_third,'o',x,x,'r',x,slope*x,'g')
[27] text('$H_{m_0}$ [m]', 'x')
[28] text('$H_{1/3}$ [m]', 'y')
[29] str = sprintf('%0.3f~H_{m_0}$',slope);
[30] legend({'$H_{1/3}$','$H_{m_0}$',str},'interpreter','latex')
[31] legend('location','northwest')
[32] graphToPDF('swH.pdf',4,4)
[33] end
[34]
[35] function H_third = comp_H_third(eta)
[36] ind_cross = find(diff(eta>0)==(-1)); % Find downward crossing points.
[37] N = length(ind_cross)-1; % The number of waves.
[38] H = NaN(1,N); % Allocate memory for wave heights.
[39] for n = 1:N
[40] rng = (ind_cross(n)+1):ind_cross(n+1); % One wave section of data.
[41] H(n) = max(eta(rng)) - min(eta(rng)); % Compute wave height.
[42] end
[43] [~,ind] = sort(H,'descend');
[44] H = H(ind); % Put in descending order.
[45] H_third = mean(H(1:round(N/3))); % Compute H_1/3.
[46] end
```



We see that the  $H_{m_0}$  overestimates  $H_{1/3}$  by about ten percent which is in line with figure 4.11 from the textbook.

7.3 Let  $u = -\frac{5}{4} \frac{f_{\text{PM}}^4}{f^4}$  then  $du = 5 \frac{f_{\text{PM}}^4}{f^5} df$  so  $\frac{1}{5f_{\text{PM}}^4} du = \frac{1}{f^5} df$ .

$$\begin{aligned} [\eta^2] &= m_0 = \int_0^\infty E(f) df \\ &= \int_0^\infty \alpha_{\text{PM}} \frac{g^2}{(2\pi)^4 f^5} e^{-\frac{5}{4} \left(\frac{f_{\text{PM}}}{f}\right)^4} df \\ &= \alpha_{\text{PM}} \frac{g^2}{(2\pi)^4 5 f_{\text{PM}}^4} \int_{-\infty}^0 e^u du \\ &= \alpha_{\text{PM}} \frac{g^2}{(2\pi)^4 5 f_{\text{PM}}^4} \\ \rightarrow \alpha_{\text{PM}} &= [\eta^2] \frac{(2\pi)^4 5 f_{\text{PM}}^4}{g^2} \end{aligned}$$

and so

$$\begin{aligned} E_{\text{PM}}(f) &= [\eta^2] \frac{(2\pi)^4 5 f_{\text{PM}}^4}{g^2} \frac{g^2}{(2\pi)^4 f^5} e^{-\frac{5}{4} \left(\frac{f_{\text{PM}}}{f}\right)^4} \\ &= 5 [\eta^2] \frac{f_{\text{PM}}^4}{f^5} e^{-\frac{5}{4} \left(\frac{f_{\text{PM}}}{f}\right)^4} \end{aligned}$$

## 7.4

$$\begin{aligned} \log [E_{\text{PM}}] &= \log [5 [\eta^2] f_{\text{PM}}^4] - 5 \log f - \frac{5}{4} \frac{f_{\text{PM}}^4}{f^4} \\ \frac{\partial E_{\text{PM}}}{\partial f} &= \frac{\partial}{\partial f} \log [E_{\text{PM}}] = 0 - 5 \frac{1}{f} + 5 \frac{f_{\text{PM}}^4}{f^5} = 5 \frac{f_{\text{PM}}^4 - f^4}{f^5} \\ &\rightarrow \frac{\partial E_{\text{PM}}}{\partial f} = 5 \frac{f_{\text{PM}}^4 - f^4}{f^5} E_{\text{PM}} \end{aligned}$$

This maximum occurs where the slope is zero and this happens with  $f = f_{\text{PM}}$  so we see that the peak frequency is  $f_p = f_{\text{PM}}$ . Now we can write

$$E_{\text{PM}} df = 5 [\eta^2] \frac{f_{\text{PM}}^5}{f^5} e^{-\frac{5}{4} \left(\frac{f_{\text{PM}}}{f}\right)^4} \frac{df}{f_{\text{PM}}} = [\eta^2] \frac{5}{u^5} e^{-\frac{5}{4} u^4} du$$

so we see that

$$E(u) = [\eta^2] \frac{5}{u^5} e^{-\frac{5}{4} u^4}$$

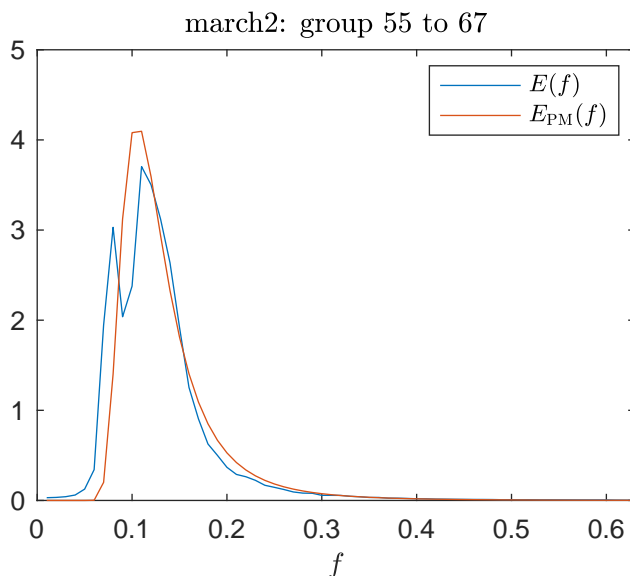
**7.5** The following code produces a graph of the data and the least squares fit of the Pierson-Moskowitz model to the data.

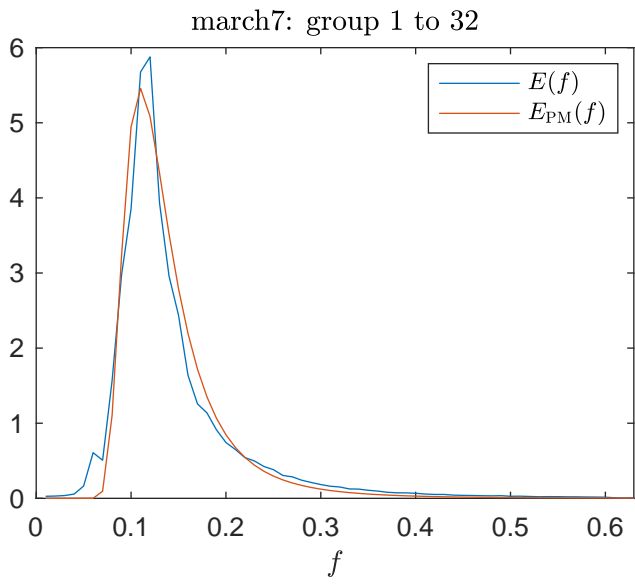
```
[1] function buoyCheckPM(day,nKeep)
[2] % Check to see how well the Pierson-Moskowitz spectrum fits the data.
[3] march2 = '.././waveData/xy029p101_201903020000-201903032359';
[4] march6 = '.././waveData/xy029p101_201903060000-201903062359';
[5] march7 = '.././waveData/xy029p101_201903070000-201903072359';
[6] march8 = '.././waveData/xy029p101_201903080000-201903082359';
[7] switch day
[8]     case 'march2'
[9]         fname = march2;
[10]    case 'march6'
[11]        fname = march6;
[12]    case 'march7'
[13]        fname = march7;
[14]    case 'march8'
[15]        fname = march8;
[16] end
[17] [r,stepError] = readBuoyFile(fname);
[18] fs = 1.28;
[19] N30 = 60*30*fs;    % number of samples in 30 minutes
[20] Ns = size(r,1);    % number of samples.
[21] Ng = floor(Ns/N30); % number of 30 minute groups of samples.
[22] Nfft = 128;        % the rest assumes N30/Nfft is an integer.
[23] Ne = N30/Nfft;     % number of FFT's in a 30 minute ensemble.
[24] eta = reshape(r(1:Ng*N30,3)/100, Nfft, Ne , Ng);
[25] error = sum(reshape(stepError(1:Ng*N30) , N30, Ng));
[26] %eta = eta(:,:,error==0);    % Keep just the error free groups.
[27] %Ng = size(eta,3);
[28] eta = eta - mean(eta,1);    % Remove mean from each column.
[29] En = NaN(Nfft/2,Ng);    % Set up array for spectral density.
[30] df = fs/Nfft;    % Compute the frequency spacing.
[31] for ng = 1:Ng    % Compute spectral density.
[32]     if ~error(ng)
[33]         H = fft(squeeze(eta(:,:,ng))); % Compute FFT of sections in group.
[34]         H = H(1:Nfft/2,:); % Cut off the redundant half.
[35]         epsilon = (2/(Nfft*fs))*abs(H).^2;
[36]         En(:,ng) = mean(epsilon,2); % Density is the mean of epsilon.
[37]     end
[38] end
[39] En = En(2:end,:);
[40] fn = (1:(Nfft/2-1))*df;
[41] m0 = sum(En,1)*df;
[42] %m2 = sum(fn.^2.*En,1)*df;
[43] logE = log(En./m0);
[44] figure(1)
[45] surf(logE)
[46] zlim([-6,6])
[47]
[48] if nargin == 1
[49]     nKeep = 1:size(En,2);
[50] end
[51] E = mean(En(:,nKeep),2,'omitnan');
[52] mm0 = mean(m0(nKeep),'omitnan');
```

```

[53] [~,ind] = max(E);
[54] fp_0 = fn(ind);
[55] fp_fit = fit_Epm(fn,E,mm0,fp_0);
[56] Epm_fit = PM_spec(fn,fp_fit,mm0);
[57]
[58] figure(2)
[59] plot(fn,E,fn,Epm_fit);
[60] legend({'$E(f)$', '$E_{\rm PM}(f)$'}, 'interpreter', 'latex')
[61] texText('$f$', 'x')
[62] texText(sprintf('%s: group %d to %d', day, min(nKeep), max(nKeep)), 't')
[63] graphToPDF(sprintf('PM-fit-%s.pdf', day), 4, 3);
[64]
[65] end
[66]
[67] function Epm = PM_spec(f,fp,m0)
[68]     u = fp./f;
[69]     Epm = (5*m0/fp)*u.^5.*exp(-1.25*u.^4);
[70] end
[71]
[72] function fp_fit = fit_Epm(f,E,m0,fp_0)
[73] % This function finds the Pierson-Moskowitz spectrum that minimizes the
[74] % mean squared error between the observed data (f,E) and the model. This is
[75] % accomplished by finding that parameter fp that minimizes the error.
[76]     fp_fit = fminsearch(@compMSE,fp_0);
[77]     function mse = compMSE(fp)
[78]         E_fit = PM_spec(f,fp,m0);
[79]         mse = sum((E-E_fit).^2);
[80]     end
[81] end

```





**7.6** The following code makes the graph. The model only seems to fit the data very well for March 6.

```
[1] function buoyCheckJS(day,nKeep)
[2] % Check to see how well the Pierson-Moskowitz spectrum fits the data.
[3] march2 = '.././waveData/xy029p101_201903020000-201903032359';
[4] march6 = '.././waveData/xy029p101_201903060000-201903062359';
[5] march7 = '.././waveData/xy029p101_201903070000-201903072359';
[6] march8 = '.././waveData/xy029p101_201903080000-201903082359';
[7] switch day
[8]     case 'march2'
[9]         fname = march2;
[10]    case 'march6'
[11]        fname = march6;
[12]    case 'march7'
[13]        fname = march7;
[14]    case 'march8'
[15]        fname = march8;
[16] end
[17] [r,stepError] = readBuoyFile(fname);
[18] fs = 1.28;
[19] N30 = 60*30*fs;      % number of samples in 30 minutes
[20] Ns = size(r,1);     % number of samples.
[21] Ng = floor(Ns/N30); % number of 30 minute groups of samples.
[22] Nfft = 128;        % the rest assumes N30/Nfft is an integer.
[23] Ne = N30/Nfft;    % number of FFT's in a 30 minute ensemble.
[24] eta = reshape(r(1:Ng*N30,3)/100, Nfft, Ne , Ng);
[25] error = sum(reshape(stepError(1:Ng*N30) , N30, Ng));
[26] %eta = eta(:,:,error==0);      % Keep just the error free groups.
[27] %Ng = size(eta,3);
[28] eta = eta - mean(eta,1);      % Remove mean from each column.
[29] En = NaN(Nfft/2,Ng);        % Set up array for spectral density.
[30] df = fs/Nfft;              % Compute the frequency spacing.
[31] for ng = 1:Ng              % Compute spectral density.
[32]     if ~error(ng)
[33]         H = fft(squeeze(eta(:,:,ng))); % Compute FFT of sections in group.
[34]         H = H(1:Nfft/2,:); % Cut off the redundant half.
[35]         epsilon = (2/(Nfft*fs))*abs(H).^2;
[36]         En(:,ng) = mean(epsilon,2); % Density is the mean of epsilon.
[37]     end
[38] end
[39] En = En(2:end,:);
[40] fn = (1:(Nfft/2-1))*df;
[41] m0 = sum(En,1)*df;
[42] %m2 = sum(fn.^2.*En,1)*df;
[43] logE = log(En./m0);
[44] figure(1)
[45] surf(logE)
[46] zlim([-6,6])
[47]
[48] if nargin == 1
[49]     nKeep = 1:size(En,2);
[50] end
[51] E = mean(En(:,nKeep),2,'omitnan');
[52] mm0 = mean(m0(nKeep) , 'omitnan');
```



```

[53] [~,ind] = max(E);
[54]
[55] % initial guesses
[56] fp_0     = fn(ind);
[57] alpha    = 5*mm0*(2*pi*fp_0)^4/9.8^2 ;
[58] sigma_a  = 0.07;
[59] sigma_b  = 0.09;
[60] gamma    = 3.3;
[61] par_0    = [alpha,fp_0,sigma_a,sigma_b,gamma];
[62] [par_fit,E_fit,f_fit] = fit_JONSWAP(fn,E,par_0);
[63]
[64] figure(2)
[65] plot(fn,E,f_fit,E_fit);
[66] legend({'$E(f)$', '$E_{\rm JS}(f)$'}, 'interpreter', 'latex')
[67] texText('$f$', 'x')
[68] texText(sprintf('%s: group %d to %d', day, min(nKeep), max(nKeep)), 't')
[69] parName = {'\alpha', 'f_p', '\sigma_a', '\sigma_b', '\gamma'};
[70] str = '';
[71] for n = 1:5
[72]     str = sprintf('%s%s=%0.3f$\n', str, parName{n}, par_fit(n));
[73] end
[74] texText(str, 'g', 0.7*max(fn), 0.5*max(E))
[75] graphToPDF(sprintf('JS-fit-%s.pdf', day), 4, 3);
[76]
[77] end
[78]

```

```

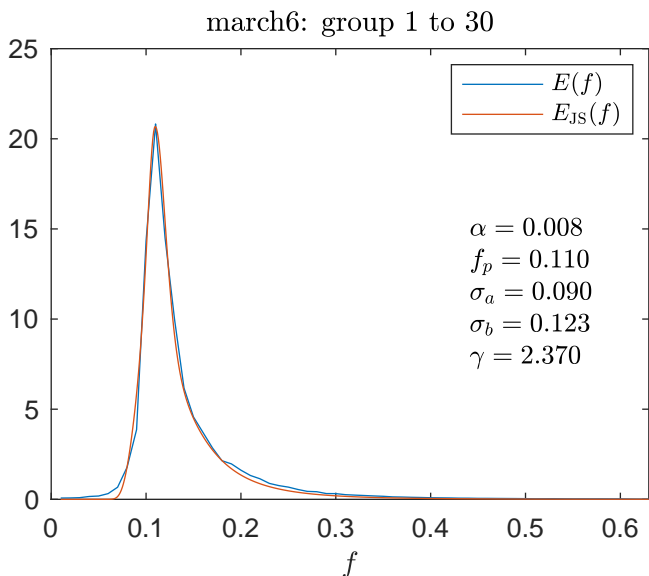
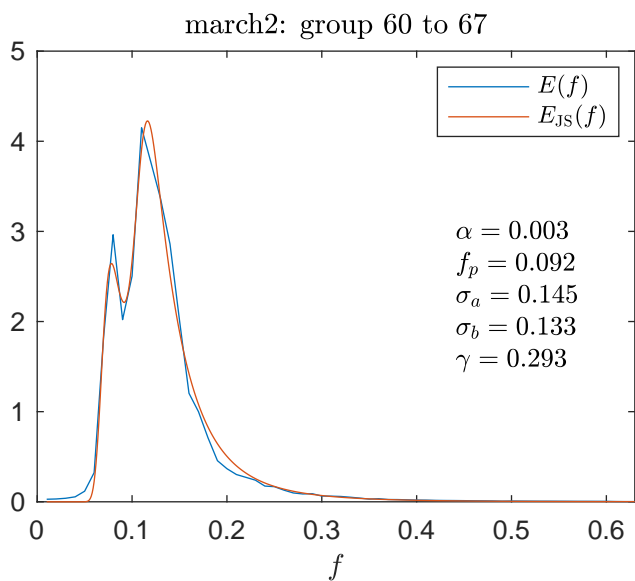
[1] function [par_fit,fit_func,f_fine] = fit_JONSWAP(f,E,par_0)
[2] % This function finds the Pierson-Moskowitz spectrum that minimizes the
[3] % mean squared error between the observed data (f,E) and the model. This is
[4] % accomplished by finding that parameter fp that minimizes the erro.
[5]     par_fit = fminsearch(@compSSE,par_0);
[6]     df = f(2)-f(1);
[7]     f_fine = min(f):df/10:max(f);
[8]     fit_func = JS_spec(f_fine,par_fit);
[9]     function mse = compSSE(par)
[10] %         if sum(par>0)==5
[11] %             E_fit = JS_spec(f,par);
[12] %             mse = sum((E-E_fit).^2);
[13] %         else
[14] %             mse = Inf;
[15] %         end
[16] %         E_fit = JS_spec(f,par);
[17] %         mse = sum((E-E_fit).^2);
[18] %         if sum(par>0)<5 || par(3)<0.03
[19] %             mse = 1.1*mse;
[20] %         end
[21]
[22]     end
[23] end
[24]
[25] function E_J = JS_spec(f,par) % JONSWAP spectral density
[26]     alpha    = par(1);

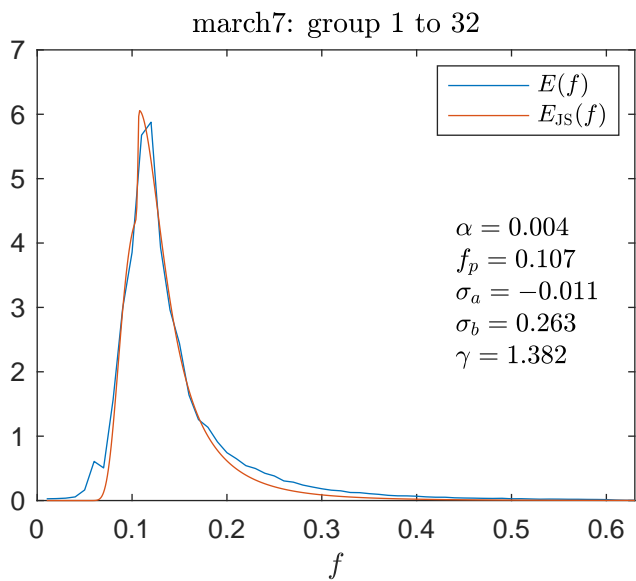
```

```

[27] fp      = par(2);
[28] sigma_a = par(3);
[29] sigma_b = par(4);
[30] gamma   = par(5);
[31] g = 9.8;
[32]
[33] top = f>fp; % one where f> fp zero otherwise;
[34] bot = ~top; % one where f<=fp zero otherwise;
[35] sigma = sigma_a*bot + sigma_b*top;
[36] u = f/fp;
[37] PM = alpha*(g/(2*pi)^2)^2*f.^-5.*exp(-1.25*u.^-4);
[38] G = gamma.^(exp(-0.5*((u-1)./sigma).^2));
[39] E_J = PM.*G;
[40] end
[41]

```





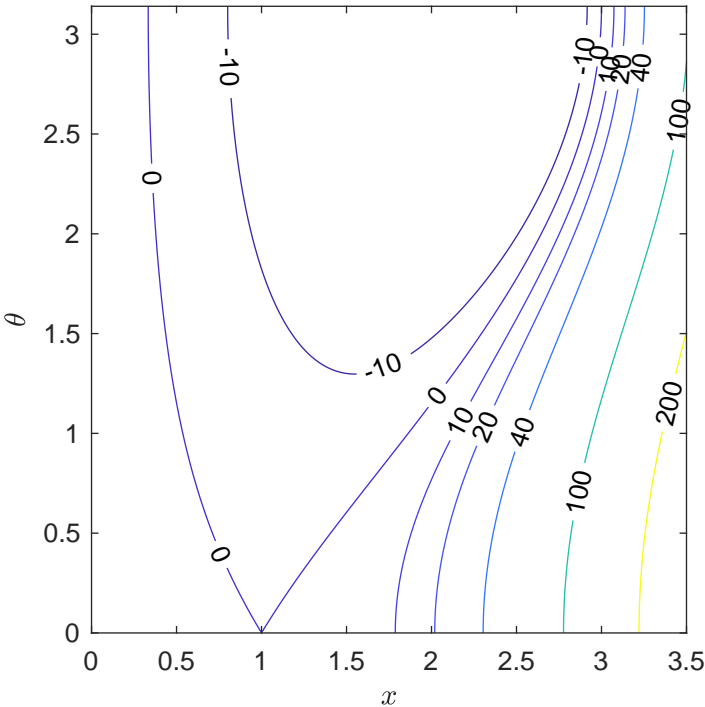
7.7

$$\begin{aligned}
 k_+^2 &= \vec{k}_+ \cdot \vec{k}_+ \\
 &= \frac{\vec{k}_1 + \vec{k}_2}{2} \cdot \frac{\vec{k}_1 + \vec{k}_2}{2} \\
 &= \frac{1}{4} [\vec{k}_1 \cdot \vec{k}_1 + \vec{k}_2 \cdot \vec{k}_2 + 2\vec{k}_1 \cdot \vec{k}_2] \\
 &= \frac{1}{4} [k_1^2 + k_2^2 + 2k_1k_2 \cos \theta] \\
 \rightarrow g^2 k_+^2 &= \frac{1}{4} [g^2 k_1^2 + g^2 k_2^2 + 2gk_1gk_2 \cos \theta] \\
 &= \frac{1}{4} [\omega_1^4 + \omega_2^4 + 2\omega_1^2\omega_2^2 \cos \theta]
 \end{aligned}$$

So  $\omega_+^2 = gk_+$  if  $\omega_+^4 = g^2 k_+^2$  or

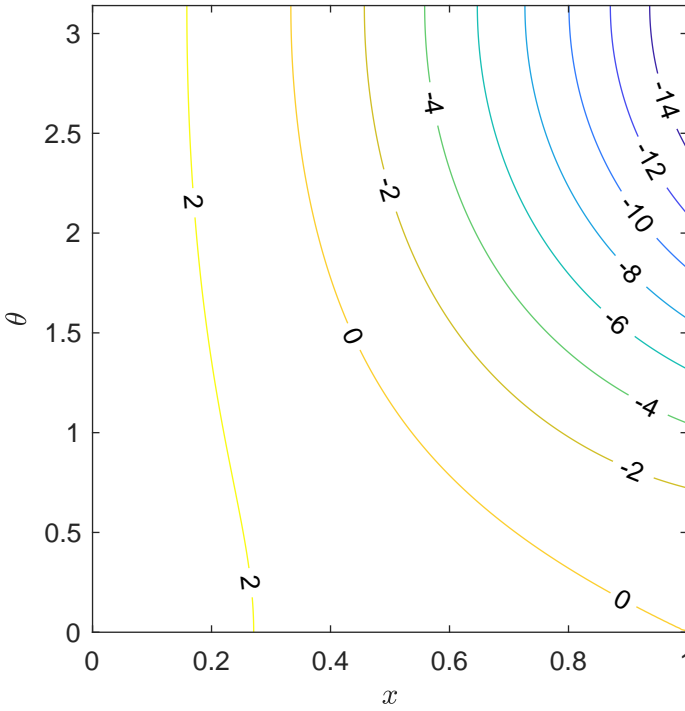
$$\begin{aligned}
 0 &= g^2 k_+^2 - \omega_+^4 \\
 &= g^2 k_+^2 - \left(\frac{\omega_1 + \omega_2}{2}\right)^4 \\
 \rightarrow 0 &= 16g^2 k_+^2 - (\omega_1 + \omega_2)^4 \\
 \rightarrow 0 &= 16 \frac{1}{4} [\omega_1^4 + \omega_2^4 + 2\omega_1^2\omega_2^2 \cos \theta] - (\omega_1 + \omega_2)^4 \\
 \rightarrow 0 &= 4 [\omega_1^4 + \omega_2^4 + 2\omega_1^2\omega_2^2 \cos \theta] - (\omega_1 + \omega_2)^4 \\
 \rightarrow \frac{0}{\omega_2^4} &= 4 [x^4 + 1 + 2x^2 \cos \theta] - (x + 1)^4 \\
 \rightarrow 0 &= 4 [x^4 + 1 + 2x^2 \cos \theta] - (x + 1)^4
 \end{aligned}$$

with  $x = \frac{\omega_1}{\omega_2}$ . If we make a contour graph of the two dimensional function  $4 [x^4 + 1 + 2x^2 \cos \theta] - (x + 1)^4$ , we see that there is two zeros to the function for each value of  $\theta$ .



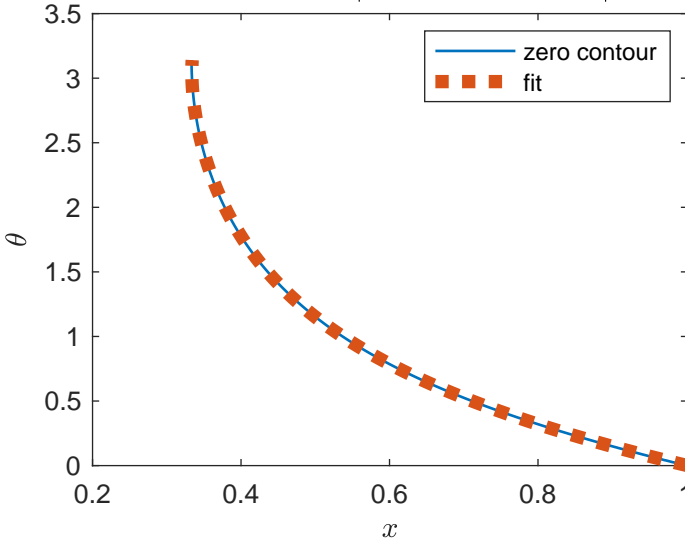
It appears that for a given  $\theta$  the two solutions are inverses of each other. This appears to be

true algebraically: suppose that  $x$  is such that  $0 = 4 [x^4 + 1 + 2x^2 \cos \theta] - (x + 1)^4$ , then if we divide by  $x^4$  we have  $0 = 4 [1 + \frac{1}{x^4} + 2\frac{1}{x^2} \cos \theta] - (1 + \frac{1}{x})^4$ , now let  $y = 1/x$ , and substitute to get  $0 = 4 [1 + y^4 + 2y^2 \cos \theta] - (1 + y)^4$  which is the same equation as before, so we see that  $y = 1/x$  is a solution if  $x$  is. So when we look for a solution we can just look for the one that is less than 1.



Unexpectedly the zero contour is fit well by a fourth order polynomial in  $\theta$ .

$$0.0078\theta^4 - 0.0790\theta^3 + 0.335\theta^2 - 0.729\theta + 1.001$$



7.8