Braidan Duffy

OCE3521

Homework 2

Problem 1 – DD 3.3

The horizontal velocity of a fluid particle is defined by

$$u = \frac{d\xi}{\mathrm{dt}} = 40\,\frac{\mathrm{cm}}{\mathrm{s}} \qquad [1.1]$$

Integrating the horizontal velocity yields the horizontal position of the fluid particle over time

$$\xi = \int u dt = 40t = \mathrm{x} \qquad [1.2]$$

The x-function from Equation 1.2 can be introduced to the boundary function definition showing how the fluid particle's vertical position changes over time

$$\zeta = 30e^{-0.02x} = 30e^{-0.02(40t)} = 30e^{-0.8t} \qquad [1.3]$$

The particle's vertical velocity along the boundary can then be determined by

$$w = \frac{d\zeta}{dt} = -24e^{-0.8t} \qquad [1.4]$$

By plugging in $x = 50$ cm to Equation 1.2, the time it takes the fluid particle to travel 50 cm from the origin and meet the boundary is $t = 1.25$ s. Taking that value into Equation 1.4 yields the fluid particle's vertical velocity at $x = 50$ cm

$$w = -24e^{-0.8(1.25)} = -24e^{-1} = -8.83\,\frac{\mathrm{cm}}{\mathrm{s}} \qquad [1.5]$$

Problem 2 – DD 3.4

Part A)

$$\text{Upper bound: } \zeta(x,0) = 30e^{-0.02x} \qquad [2.1]$$

$$\text{Lower bound: } \zeta(\mathrm{x},0) = 0 \qquad [2.2]$$

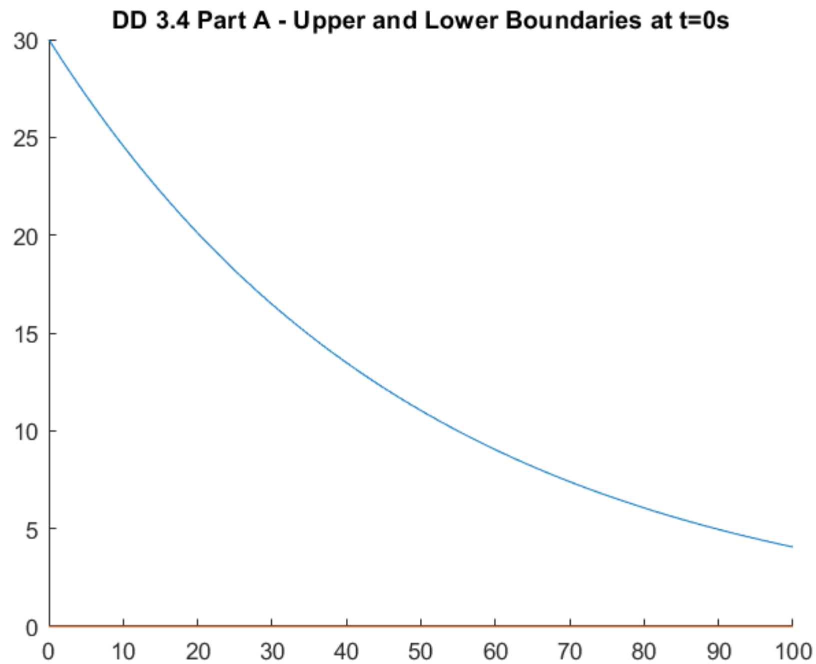

Figure 1: Plot of the bounds as defined by Equations 2.1 and 2.2

Part B)

For Part B, a simplification was made to ease calculation. Since the upper bound exponentially decays to 0, the rate of change at x=0 was determined and used to define how the boundary changes over time.

$$\frac{\partial \zeta}{\partial t}_{x=0} = 3e^{0.1t} > 0 \qquad [2.3]$$

Since $\frac{\partial \zeta}{\partial t}_{x=0}$ will always be greater than it can be surmised that the z-intercept of the upper boundary will increase over time. Since the function remains exponentially decaying to 0, as the function approaches infinity, the bound does not change.

Part C)

At t=10 s, the upper boundary function is defined by

$$\zeta(x, 10) = 30e^{-0.02x+1} \qquad [2.4]$$

Solving for *w* uses the same process as shown in Problem 1 – DD 3.3 Equations 1.3-1.5. For this problem however, the time derivative of the boundary function is

$$w = \frac{d\zeta}{dt} = -24e^{-0.8t+1}$$

Which, when $x = 50$ cm, yields the particles vertical velocity as

$$w = -24e^{-0.8(1.25)+1} = -24e^{0} = -24 \text{ cm/s} \qquad [2.5]$$

Problem 3 – DD 3.7

To begin this problem, we must start with the dispersion equation

$$\left(\frac{2\pi}{T}\right)^2 = \frac{g2\pi}{L}\tanh\left(\frac{2\pi h}{L}\right) \tag{3.1}$$

Several variables cancel out and we can rearrange the equation to the following

$$\frac{2\pi L}{gT^2} = \tanh\left(\frac{2\pi h}{L}\right) \tag{3.2}$$

The deep-water wavelength is defined as $L_0 = \frac{gT^2}{2\pi}$ which can be substituted in Equation 3.2 yielding

$$\frac{L}{L_0} = \tanh\left(\frac{2\pi h}{L}\right) \tag{3.3}$$

This can then be rearranged to solve for *h* which yields the final answer

$$h = \frac{L}{2\pi}\operatorname{arctanh}\left(\frac{L}{L_0}\right) = \frac{200}{2\pi}\operatorname{arctanh}\left(\frac{200}{312}\right) = 24.189\text{ m} \tag{3.4}$$

Problem 4 – DD 3.9

To solve this problem several assumptions are made. Firstly, the fluid is assumed to be incompressible and irrotational as well as initially at rest. Since the bottom is sloped, the theory of relative triangles can be used to determine the equation for the water depth along the x-axis

$$y = mx - h_0 \tag{4.1}$$

At $t = 0, x = 0, z = -h_0$. Since the bottom boundary is oscillating over time, the water depth becomes a function of the bottom slope, initial depth, and bottom boundary movement

$$y = mx - h_0 + \zeta(x.z.t) \tag{4.2}$$

Assuming the bottom boundary has all the properties necessary to complete a Fourier and Laplace transform. The velocity potential through this problem is $\phi(x, z, t)$ which allows the fluid velocity vector to be defined as

$$q = \nabla\phi \tag{4.3}$$

Equation 4.3 forces the continuity equation to become

$$\nabla q = \nabla\phi = 0, (x, z, t) \in \mathrm{D} \tag{4.4}$$

Where D is the fluid domain in the *x*, *z*, and *t* dimensions. Equation 4.4 must satisfy the kinematic boundary conditions at the free surface and solid boundary defined by

$$\frac{\partial\phi}{\partial z} = \frac{\partial\eta}{\partial t} + \frac{\partial\phi}{\partial x}\frac{\partial\eta}{\partial x} + \frac{\partial\phi}{\partial z}\frac{\partial\eta}{\partial y}\ ; y = \eta(x, z, t) \tag{4.5}$$

$$\frac{\partial \phi}{\partial z} = \frac{\partial \zeta}{\partial t} + \frac{\partial \phi}{\partial x}\frac{\partial \zeta}{\partial x} + \frac{\partial \phi}{\partial z}\frac{\partial \zeta}{\partial y} \; ; y = -(mx - h_0) + \zeta(x, z, t) \qquad [4.6]$$

The conditions specified above are only valid if the viscous and capillary effects are negligible. The dynamic condition at the free surface can then be determined by

$$\frac{\partial \phi}{\partial z} + \frac{1}{2}|\nabla \phi 2| + g\eta = 0 \; ; y = \eta(x, z, t) \qquad [4.7]$$

At $t = 0$ seconds, the initial conditions given by $\eta(x, z, t) = 0$ and $\eta(x, z, 0)$, making new independent variables

$$x = kx, z = kz, y = ky, \text{and } t = \sigma t$$

Combining these variables in Equation 4.4 yields a new dynamic boundary condition

$$\Delta \phi = 0, (x, z, t) \in \mathrm{D} \qquad [4.8]$$

Plugging Equation 4.8 into Equations 4.5 and 4.6 yields

$$\frac{\partial \phi}{\partial y} = \frac{\partial \eta}{\partial t} + ka\left(\frac{\partial \phi}{\partial x}\frac{\partial \eta}{\partial x} + \frac{\partial \phi}{\partial z}\frac{\partial \eta}{\partial z}\right); y = ka\eta(x, z, t) \qquad [4.9]$$

$$\frac{\partial \phi}{\partial z} = \frac{\partial \eta}{\partial t} + ka\left(\frac{\partial \phi}{\partial x}\frac{\partial \eta}{\partial x} + \frac{\partial \phi}{\partial z}\frac{\partial \eta}{\partial z}\right); y = -(mx - h_0) + kz\zeta(x, z, t) \qquad [4.10]$$

$$\rightarrow \frac{\partial \phi}{\partial y} = \frac{1}{2} + kz|\nabla \phi|^2 + \frac{gk}{\sigma^2} = 0 \; ; y = ka\eta(x, z, t) \qquad [4.11]$$

To solve this BVP, the conditions must be linearized using the method of linear transforms (e.g. Laplace of Fourier transforms).

Problem 5 – DD 3.11

This problem presents an issue at the start since the wavenumber, k, is not given. It must therefore be calculated via a programmable calculator. In this case, Problem 8 – DD 3.15 already requires a calculator be written, so that same code can be used for this problem to get k. The stream function can then be rearranged to solve for x, since it is the simpler variable to isolate

$$\psi = -\frac{Hg}{2\sigma}\frac{\sinh\big(k(h + z)\big)}{\cosh(kh)}\cos(kx - \sigma t) \qquad [5.1]$$

At $t = 0$ seconds

$$-\frac{HgT}{4\pi}\frac{\sinh\big(k(h + z)\big)}{\cosh(kh)}\cos(kx) = \psi$$

$$-\frac{4\pi \cosh(kh)\,\psi}{HgT sinh\big(k(h + z)\big)} = \cos(kx)$$

$$k \arccos\left(-\frac{4\pi \cosh(kh)\,\psi}{HgT \sinh\big(k(h+z)\big)}\right) = x \qquad [5.2]$$

This equation was programmed and plotted in MATLAB 2020a (See Appendix A – OCE3521_Homework2_Duffy.m). When ψ is set to -1, 0, and 1, the following figure is produced

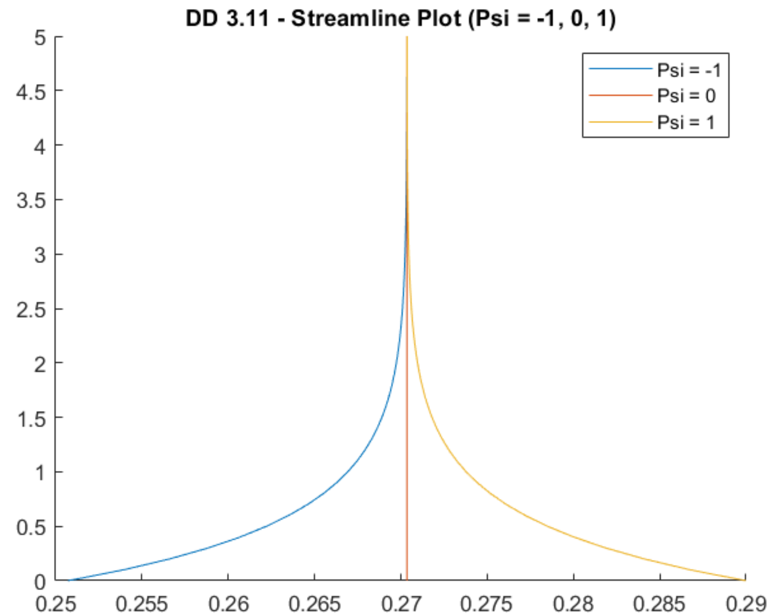

Figure 2: Plot of Equation 5.2 for Psi = -1, 0, 1

Problem 6 – DD 3.12

To begin this problem, the wave period must first be calculated. Since the wave is in deep water, we can use the deep-water approximation to get

$$T = \sqrt{\frac{L_0 2\pi}{g}} = \sqrt{\frac{200(2\pi)}{9.81}} = 11.32 \text{ s} \qquad [6.1]$$

With the wave period known, we can determine the wave celerity as

$$C = \frac{L_0}{T} = \frac{200}{11.32} = 17.67 \frac{\text{m}}{\text{s}} \qquad [6.2]$$

Since this problem involves to distinct bodies moving in the same direction, the relative velocity can be used to determine the ship's velocity

$$u_r = u_s - C \qquad [6.3]$$

The relative velocity of the ship and wave is defined simply by

$$u_r = \frac{L_{ship}}{t_{overtake}} = \frac{100}{20} = 5\,\frac{\mathrm{m}}{\mathrm{s}} \quad [6.4]$$

Rearranging Equation 6.3 to solve for u_s yields the final answer

$$u_s = u_r + C = 5 + 17.67 = 22.67\,\frac{\mathrm{m}}{\mathrm{s}} \quad [6.5]$$

Problem 7 – DD 3.13

To begin the problem, the theory of relative triangles is used to generate an expression for water depth in terms of x (distance from the shore). Solving it will also determine the distance from the sensor to the shore.

$$\frac{h}{x} = \frac{.005}{1} \xrightarrow{yields} x = \frac{h}{0.005} \xRightarrow{h=200\text{ m}} \frac{200}{0.005} = 40{,}000\text{ m} \quad [7.1]$$

Since the tsunami has an infinite wavelength, it can be modeled as a shallow-water wave over any water depth, h. The celerity for the tsunami in terms of x is given by

$$C = \sqrt{gh} = \sqrt{0.005gx} \quad [7.2]$$

The time the tsunami takes to travel from the sensor head to the shoreline is determined by the integral of the celerity over the distance to the shore

$$dt = \frac{dx}{C(x)} \xrightarrow{yields} t = \int_0^{40000} (0.005gx)^{-\frac{1}{2}} dx = 1806\text{ s} = 30.1\text{ mins} \quad [7.3]$$

Using the result from Equation 7.3, the tsunami will hit the shoreline at 12:30:06 local time.

Problem 8 – DD 3.15

```
% Calculates the wavenumber, k, for a wave using an iterative process and
% the dispersion equation. The equation for k was determined by rearraning
% the dispersion equation into
%           k = sigma^2 / (g * tanh(k*h))
% @param sigma: the angular frequency of the wave in Hz
% @param h: water depth in meters (m)
% @param g: acceleration due to gravity in m/s^2 (Defaults to 9.81)
% @param tol: the accuracy tolerance for the calculation in meters(Defaults 0.001)
% @return wavenumber: returns the wavenumber, k, as calculated from the dispersion equation
% @return num_iter: returns the number iterations it took to calculate the wavenumber
function [wavenumber, num_iter] = calculate_wavenumber_dispersion(sigma, h, g, tol)
    arguments
        sigma
        h
        g = 9.81
        tol = 0.001
    end

    num_iter = 0;
```

```
    y = 0;
    k = 100;

    while(abs(k-y) > tol) % Iterate through dispersion equation until tolerance is reached
        y = k; % Save previous iteration of L
        k = sigma^2 / (g * tanh(k*h)); % Calculate new value of k
        L = abs((k+y) / 2); % Find average between calculations to get towards tolerance
        num_iter = num_iter + 1; % Increment number of iterations
    end
    wavenumber = k; % Return the final L value as the theoretical wavelength
end
```

Published with MATLAB® R2020b

Appendix A – OCE3521_Homework2_Duffy.m

Problem 2 - DD 3.4

The equation for the upper moving boundary of a fluid is Zeta(x, t) = 30e^(-(0.02x-0.1t)) The lower boundary is expressed by Zeta(x, t) = 0 a) Sketch the boundaries for t=0 s

```
x_vector = 0:0.1:100; % Generate x values
upper_bound = 30 * exp(-0.02 .* x_vector); % Generate upper boundary

figure(1)
hold on
plot(x_vector, upper_bound)
plot(x_vector, zeros([1 length(x_vector)]))
hold off
title("DD 3.4 Part A - Upper and Lower Boundaries at t=0s")

% b) Discuss the motional characteristics of the upper boundary
```

Problem 5 - DD 3.11

Draw the streamlines for the following function psi = H/2*g/sigma*sinh(k(h+z))/cosh(kh)*cos(kx-sigma*t)

```
T = 5; % s
h = 10; % m
H = 2; % m
g = 9.81; % m/s^2
k = calculate_wavenumber_dispersion(2*pi/T, h); % caluclates k iteratively using dispersion
equation
psi = -1:1;
z = 0:0.1:15;

% Plot streamlines
figure(2)
hold on
for N=1:length(psi)
```

```
    for z=1:length(Z)
        X(z) = k*acos(-4*pi*cosh(k*h)*psi(N)/(H*g*T*sinh(k*(h+z))));
    end
    plot(X, Z)
end
hold off
title("DD 3.11 - Streamline Plot (Psi = -1, 0, 1)")
legend("Psi = -1", "Psi = 0", "Psi = 1")
ylim([0 5])
```

Problem 8 - DD 3.15

Develop and iterative technique to solve the dispersion relationship for k five sigma and h. (sigma)^2 = gktanh(kh) => k = (sigma)^2 / (gtanh(kh))

```
% See calculate_wavenumber_dispersion.m for the code
```

Published with MATLAB® R2020a