Water wave dispersion - 09-25-16

N. T. Gladd

Initialization: Be sure the files *NTGStylesheet2.nb* and *NTGUtilityFunctions.m* is are in the same directory as that from which this notebook was loaded. Then execute the cell immediately below by mousing left on the cell bar to the right of that cell and then typing “shift” + “enter”. Respond “Yes” in response to the query to evaluate initialization cells.

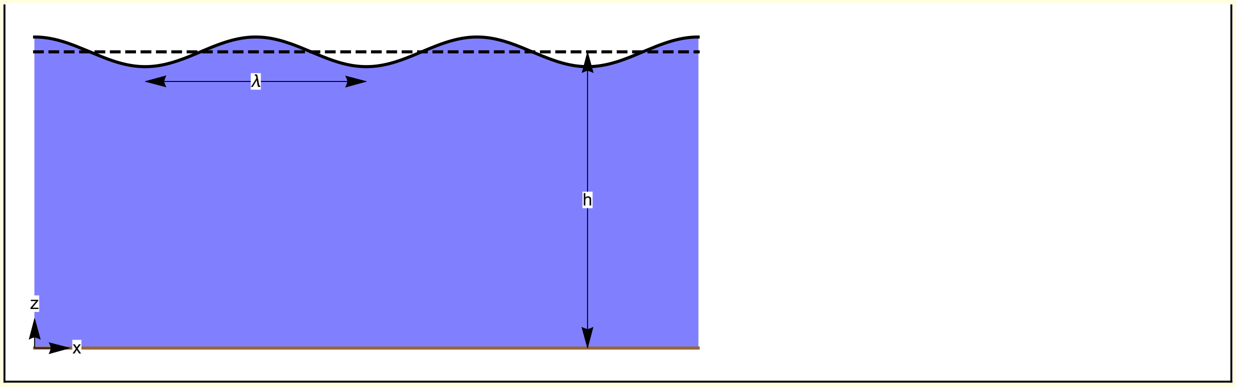
In[7]:=
```
SetDirectory[NotebookDirectory[]];
(* set directory where source files are located *)
SetOptions[EvaluationNotebook[], (* load the StyleSheet *)
 StyleDefinitions → Get["NTGStylesheet2.nb"]];
Get["NTGUtilityFunctions.m"]; (* Load utilities package *)
```

Purpose

I work through the derivation of the dispersion relation for water waves. This derivation is discussed in many places, but I will generally follow *A Modern Introduction to the Mathematics of Water Waves,* R. S. Johnson, Section 2.1. However, I will use my own notation. I start from the fluid equations derived in *Water wave equations 09-25-16*

Contents

1 Derivation of dispersion relation for water waves driven by gravity



From notebook *Water wave equations 09-25-16*, --w[16][“final”], the equations appropriate for water waves driven by gravity are

In[9]:=

```
w[1] = {V[z]^(0,0,1,0)[X, Y, Z, T] + V[y]^(0,1,0,0)[X, Y, Z, T] + V[x]^(1,0,0,0)[X, Y, Z, T] == 0,
   V[x]^(0,0,0,1)[X, Y, Z, T] + P^(1,0,0,0)[X, Y, Z, T] +
     V[x][X, Y, Z, T] V[x]^(1,0,0,0)[X, Y, Z, T] == 0, V[y]^(0,0,0,1)[X, Y, Z, T] +
     P^(0,1,0,0)[X, Y, Z, T] + V[y][X, Y, Z, T] V[y]^(0,1,0,0)[X, Y, Z, T] == 0,
   1 + δ^2 V[z]^(0,0,0,1)[X, Y, Z, T] + P^(0,0,1,0)[X, Y, Z, T] + δ^2 V[z][X, Y, Z, T]
      V[z]^(0,0,1,0)[X, Y, Z, T] == 0, V[z][X, Y, B[X], T] == V[x][X, Y, B[X], T] B′[X],
   V[z][X, Y, 1, T] == H^(0,0,1)[X, Y, T] + V[y][X, Y, 1, T] H^(0,1,0)[X, Y, T] +
     V[x][X, Y, 1, T] H^(1,0,0)[X, Y, T], P[X, Y, 1, T] == H[X, Y, T]};
w[1] = StandardizeEqn /@
  w[
   1]
```

Out[10]=

```
{V[z]^(0,0,1,0)[X, Y, Z, T] + V[y]^(0,1,0,0)[X, Y, Z, T] + V[x]^(1,0,0,0)[X, Y, Z, T] == 0,
 V[x]^(0,0,0,1)[X, Y, Z, T] + P^(1,0,0,0)[X, Y, Z, T] + V[x][X, Y, Z, T] V[x]^(1,0,0,0)[X, Y, Z, T] ==
  0, V[y]^(0,0,0,1)[X, Y, Z, T] + P^(0,1,0,0)[X, Y, Z, T] +
   V[y][X, Y, Z, T] V[y]^(0,1,0,0)[X, Y, Z, T] == 0, 1 + δ^2 V[z]^(0,0,0,1)[X, Y, Z, T] +
   P^(0,0,1,0)[X, Y, Z, T] + δ^2 V[z][X, Y, Z, T] V[z]^(0,0,1,0)[X, Y, Z, T] == 0,
 V[z][X, Y, B[X], T] - V[x][X, Y, B[X], T] B′[X] == 0,
 V[z][X, Y, 1, T] - H^(0,0,1)[X, Y, T] - V[y][X, Y, 1, T] H^(0,1,0)[X, Y, T] -
   V[x][X, Y, 1, T] H^(1,0,0)[X, Y, T] == 0, -H[X, Y, T] + P[X, Y, 1, T] == 0}
```

In[11]:=
```
DisplayEqns[w[1], {}]
```

Out[11]//TraditionalForm=

$$\begin{pmatrix} 1 & \frac{\partial V(x)(X,Y,Z,T)}{\partial X} + \frac{\partial V(y)(X,Y,Z,T)}{\partial Y} + \frac{\partial V(z)(X,Y,Z,T)}{\partial Z} = 0 \\ 2 & \frac{\partial P(X,Y,Z,T)}{\partial X} + \frac{\partial V(x)(X,Y,Z,T)}{\partial T} + \frac{\partial V(x)(X,Y,Z,T)}{\partial X} V(x)(X, Y, Z, T) = 0 \\ 3 & \frac{\partial P(X,Y,Z,T)}{\partial Y} + \frac{\partial V(y)(X,Y,Z,T)}{\partial T} + \frac{\partial V(y)(X,Y,Z,T)}{\partial Y} V(y)(X, Y, Z, T) = 0 \\ 4 & \frac{\partial V(z)(X,Y,Z,T)}{\partial T} \delta^2 + \frac{\partial V(z)(X,Y,Z,T)}{\partial Z} V(z)(X, Y, Z, T)\, \delta^2 + \frac{\partial P(X,Y,Z,T)}{\partial Z} + 1 = 0 \\ 5 & V(z)(X, Y, B(X), T) - \frac{\partial B(X)}{\partial X} V(x)(X, Y, B(X), T) = 0 \\ 6 & -\frac{\partial H(X,Y,T)}{\partial T} - \frac{\partial H(X,Y,T)}{\partial X} V(x)(X, Y, 1, T) - \frac{\partial H(X,Y,T)}{\partial Y} V(y)(X, Y, 1, T) + V(z)(X, Y, 1, T) = 0 \\ 7 & P(X, Y, 1, T) - H(X, Y, T) = 0 \end{pmatrix}$$

The first four equation are for the bulk fluid, the last three equation are boundary conditions. See *Water wave equations 09-25-16* for details and variable/parameter definitions. These equations were derived for a sloping bottom. However in the following I consider a flat bottom and set B(X) = B'(X) = 0.

Perturb the equations and reduce the spatial dimensions from x, y, z to x, z

In[12]:=
```
w[2] = w[1] /. {V[a_] → Function[{X, Y, Z, T}, ϵ V1[a][X, Z, T]],
    P → Function[{X, Y, Z, T}, P0[Z] + ϵ P1[X, Z, T]],
    H → Function[{X, Y, T}, 1 + ϵ H1[X, T]]} // Expand
```

Out[12]=

$$\{\epsilon\, V1[z]^{(0,1,0)}[X, Z, T] + \epsilon\, V1[x]^{(1,0,0)}[X, Z, T] == 0, \\ \epsilon\, V1[x]^{(0,0,1)}[X, Z, T] + \epsilon\, P1^{(1,0,0)}[X, Z, T] + \epsilon^2\, V1[x][X, Z, T]\, V1[x]^{(1,0,0)}[X, Z, T] == 0, \\ \epsilon\, V1[y]^{(0,0,1)}[X, Z, T] == 0,\ 1 + P0'[Z] + \delta^2\, \epsilon\, V1[z]^{(0,0,1)}[X, Z, T] + \\ \epsilon\, P1^{(0,1,0)}[X, Z, T] + \delta^2\, \epsilon^2\, V1[z][X, Z, T]\, V1[z]^{(0,1,0)}[X, Z, T] == 0, \\ \epsilon\, V1[z][X, B[X], T] - \epsilon\, V1[x][X, B[X], T]\, B'[X] == 0, \\ \epsilon\, V1[z][X, 1, T] - \epsilon\, H1^{(0,1)}[X, T] - \epsilon^2\, V1[x][X, 1, T]\, H1^{(1,0)}[X, T] == 0, \\ -1 - \epsilon\, H1[X, T] + P0[1] + \epsilon\, P1[X, 1, T] == 0\}$$

Equilibrium conditions are

In[13]:=
```
w[3] = w[2] /. ϵ → 0
```

Out[13]=

$$\{True, True, True, 1 + P0'[Z] == 0, True, True, -1 + P0[1] == 0\}$$

Linearize and apply the equilibrium conditions

In[14]:=

$$w[4] = w[2]\ /.\ \epsilon^{n\_/;\,n\geq 2} \to 0\ /.\ \epsilon \to 1\ /.\ \{P0[1] \to 1, P0'[Z] \to -1\}$$

Out[14]=

$$\{V1[z]^{(0,1,0)}[X, Z, T] + V1[x]^{(1,0,0)}[X, Z, T] == 0,\ V1[x]^{(0,0,1)}[X, Z, T] + P1^{(1,0,0)}[X, Z, T] == 0, \\ V1[y]^{(0,0,1)}[X, Z, T] == 0,\ \delta^2\, V1[z]^{(0,0,1)}[X, Z, T] + P1^{(0,1,0)}[X, Z, T] == 0, \\ V1[z][X, B[X], T] - V1[x][X, B[X], T]\, B'[X] == 0, \\ V1[z][X, 1, T] - H1^{(0,1)}[X, T] == 0,\ -H1[X, T] + P1[X, 1, T] == 0\}$$

The equation for V1[*y*] is no longer relevant

In[15]:=
```
w[5] = Drop[w[4], {3}]
```

Out[15]=
```
{V1[z]^(0,1,0)[X, Z, T] + V1[x]^(1,0,0)[X, Z, T] == 0,
 V1[x]^(0,0,1)[X, Z, T] + P1^(1,0,0)[X, Z, T] == 0, δ^2 V1[z]^(0,0,1)[X, Z, T] + P1^(0,1,0)[X, Z, T] == 0,
 V1[z][X, B[X], T] - V1[x][X, B[X], T] B′[X] == 0,
 V1[z][X, 1, T] - H1^(0,1)[X, T] == 0, -H1[X, T] + P1[X, 1, T] == 0}
```

To obtain the standard dispersion equation for water waves I assume the depth of the water is constant. I analyze the case of a sloping bottom in another notebook.

In[16]:=
```
w[6] = w[5] /. B[X] → 0 /. B'[X] → 0
```

Out[16]=
```
{V1[z]^(0,1,0)[X, Z, T] + V1[x]^(1,0,0)[X, Z, T] == 0, V1[x]^(0,0,1)[X, Z, T] + P1^(1,0,0)[X, Z, T] == 0,
 δ^2 V1[z]^(0,0,1)[X, Z, T] + P1^(0,1,0)[X, Z, T] == 0, V1[z][X, 0, T] == 0,
 V1[z][X, 1, T] - H1^(0,1)[X, T] == 0, -H1[X, T] + P1[X, 1, T] == 0}
```

There are three first order pdes and three boundary conditions

In[17]:=
```
DisplayEqns[w[6], {V1 → V₁, P1 → P₁, H1 → H₁}]
```

Out[17]//TraditionalForm=

$$\begin{pmatrix} 1 & \frac{\partial V_1(x)(X,Z,T)}{\partial X} + \frac{\partial V_1(z)(X,Z,T)}{\partial Z} = 0 \\ 2 & \frac{\partial P_1(X,Z,T)}{\partial X} + \frac{\partial V_1(x)(X,Z,T)}{\partial T} = 0 \\ 3 & \frac{\partial V_1(z)(X,Z,T)}{\partial T}\,\delta^2 + \frac{\partial P_1(X,Z,T)}{\partial Z} = 0 \\ 4 & V_1(z)(X,\,0,\,T) = 0 \\ 5 & V_1(z)(X,\,1,\,T) - \frac{\partial H_1(X,T)}{\partial T} = 0 \\ 6 & P_1(X,\,1,\,T) - H_1(X,\,T) = 0 \end{pmatrix}$$

Assume a wave-like eikonal for the X-T dependence

In[18]:=
```
Clear[Eikonal];
Eikonal[X_, T_] := Exp[I K X - I Ω T];
```

In[20]:=
```
w[7] = w[6] /. {V1[a_] → Function[{X, Z, T}, V1[a][Z] Eikonal[X, T]], P1 →
    Function[{X, Z, T}, P1[Z] Eikonal[X, T]], H1 → Function[{X, T}, H1 Eikonal[X, T]]}
```

Out[20]=

$$\left\{ i\, e^{i K X - i T \Omega} K\, V1[x][Z] + e^{i K X - i T \Omega} V1[z]'[Z] == 0,\ i\, e^{i K X - i T \Omega} K\, P1[Z] - i\, e^{i K X - i T \Omega}\,\Omega\, V1[x][Z] == 0,\ -i\, e^{i K X - i T \Omega}\,\delta^2\,\Omega\, V1[z][Z] + e^{i K X - i T \Omega} P1'[Z] == 0,\ e^{i K X - i T \Omega} V1[z][0] == 0,\ i\, e^{i K X - i T \Omega} H1\,\Omega + e^{i K X - i T \Omega} V1[z][1] == 0,\ -e^{i K X - i T \Omega} H1 + e^{i K X - i T \Omega} P1[1] == 0 \right\}$$

In[21]:=
```
w[8] =
  Table[MapEqn[(Expand[#/(I Eikonal[X, T])]) &, w[7][[i]]], {i, 1, Length[w[7]]}];
w[8] = RefineEqn[#, {V1, P1, H1, Z}] & /@ w[8]
```

Out[22]=

$$\left\{ K\, V1[x][Z] - i\, V1[z]'[Z] == 0,\ K\, P1[Z] - \Omega\, V1[x][Z] == 0,\ -\delta^2\,\Omega\, V1[z][Z] - i\, P1'[Z] == 0,\ V1[z][0] == 0,\ H1\,\Omega - i\, V1[z][1] == 0,\ H1 - P1[1] == 0 \right\}$$

In[23]:=
```
DisplayEqns[w[8], {V1 → V₁, P1 → P₁, H1 → H₁} ]
```

Out[23]//TraditionalForm=

$$\begin{pmatrix} 1 & K\,V_1(x)(Z) - i\,\frac{\partial V_1(z)(Z)}{\partial Z} = 0 \\ 2 & K\,P_1(Z) - \Omega\,V_1(x)(Z) = 0 \\ 3 & -\Omega\,V_1(z)(Z)\,\delta^2 - i\,\frac{\partial P_1(Z)}{\partial Z} = 0 \\ 4 & V_1(z)(0) = 0 \\ 5 & \Omega\,H_1 - i\,V_1(z)(1) = 0 \\ 6 & H_1 - P_1(1) = 0 \end{pmatrix}$$

Quite a few operations are required to transform this set of equations into a dispersion relation.

Notice that there are two boundary conditions on $V_1(z)$ and one boundary condition on P_1. That suggests that $V_1[x]$ should be eliminated.

In[29]:=
```
w[9] = Eliminate[w[8], V1[x][Z]] /. And → List // Reverse
```

Out[29]=
```
{Ω V1[z]′[Z] == - i K² P1[Z], P1′[Z] == i δ² Ω V1[z][Z],
 V1[z][1] == - i H1 Ω, V1[z][0] == 0, P1[1] == H1}
```

In[25]:=
```
DisplayEqns[w[9], {V1 → V₁, P1 → P₁, H1 → H₁}]
```

Out[25]//TraditionalForm=

$$\begin{pmatrix} 1 & \Omega\,\frac{\partial V_1(z)(Z)}{\partial Z} = -i\,K^2\,P_1(Z) \\ 2 & \frac{\partial P_1(Z)}{\partial Z} = i\,\delta^2\,\Omega\,V_1(z)(Z) \\ 3 & V_1(z)(1) = -i\,\Omega\,H_1 \\ 4 & V_1(z)(0) = 0 \\ 5 & P_1(1) = H_1 \end{pmatrix}$$

Notice that equation 1 affords the opportunity to apply the boundary condition equation 5

In[32]:=
```
w[10] = Join[w[9]⟦1 ;; 4⟧, {(w[9]⟦1⟧ /. Z → 1) /. Sol[w[9]⟦5⟧, P1[1]] }]
```

Out[32]=
```
{Ω V1[z]′[Z] == - i K² P1[Z], P1′[Z] == i δ² Ω V1[z][Z],
 V1[z][1] == - i H1 Ω, V1[z][0] == 0, Ω V1[z]′[1] == - i H1 K²}
```

In[33]:=
```
DisplayEqns[w[10], {V1 → V₁, P1 → P₁, H1 → H₁}]
```

Out[33]//TraditionalForm=

$$\begin{pmatrix} 1 & \Omega\,\frac{\partial V_1(z)(Z)}{\partial Z} = -i\,K^2\,P_1(Z) \\ 2 & \frac{\partial P_1(Z)}{\partial Z} = i\,\delta^2\,\Omega\,V_1(z)(Z) \\ 3 & V_1(z)(1) = -i\,\Omega\,H_1 \\ 4 & V_1(z)(0) = 0 \\ 5 & \Omega\,\frac{\partial V_1(z)(1)}{\partial 1} = -i\,K^2\,H_1 \end{pmatrix}$$

Now, with all boundary conditions expressed in terms of $V_1[z]$, the variable P_1 can be eliminated.

In[38]:=
```
w[11] = Join[{MapEqn[D[#, Z] &, w[10][[1]] ] /. Sol[w[10][[2]], P1′[Z]]}, w[10][[3 ;; 5]]];
w[11] = RefineEqn[#, {V1, P1, H1, Z}] & /@ w[11]
```

Out[39]=
$$\{K^2\,\delta^2\,\text{V1}[z][Z] - \text{V1}[z]''[Z] == 0,\ i\,\text{H1}\,\Omega + \text{V1}[z][1] == 0,\ \text{V1}[z][0] == 0,\ i\,\text{H1}\,K^2 + \Omega\,\text{V1}[z]'[1] == 0\}$$

In[40]:=
```
DisplayEqns[w[11], {V1 → V₁, P1 → P₁, H1 → H₁}]
```

Out[40]//TraditionalForm=
$$\begin{pmatrix} 1 & K^2\,\delta^2\,V_1(z)(Z) - \frac{\partial^2 V_1(z)(Z)}{\partial Z^2} = 0 \\ 2 & i\,\Omega\,H_1 + V_1(z)(1) = 0 \\ 3 & V_1(z)(0) = 0 \\ 4 & i\,H_1\,K^2 + \Omega\,\frac{\partial V_1(z)(1)}{\partial 1} = 0 \end{pmatrix}$$

The second order ode in V_1[z] can be solved and the boundary conditions applied.

In[41]:=
```
w[12] = DSolve[w[11][[1]] , V1[z][Z], Z][[1, 1]]
```

Out[41]=
$$\text{V1}[z][Z] \to e^{K\,Z\,\delta}\,C[1] + e^{-K\,Z\,\delta}\,C[2]$$

The equations for the constants of integration are

In[42]:=
```
w[13] = {w[12] /. Z → 0, w[12] /. Z → 1} // RE
```

Out[42]=
$$\{\text{V1}[z][0] == C[1] + C[2],\ \text{V1}[z][1] == e^{K\,\delta}\,C[1] + e^{-K\,\delta}\,C[2]\}$$

Apply the boundary conditions

In[44]:=
```
w[14] = w[13] /. Sol[w[11][[3]] , V1[z][0]] /. Sol[w[11][[2]] , V1[z][1]]
```

Out[44]=
$$\{0 == C[1] + C[2],\ -i\,\text{H1}\,\Omega == e^{K\,\delta}\,C[1] + e^{-K\,\delta}\,C[2]\}$$

In[45]:=
```
w[15] = Solve[w[14], {C[1], C[2]}][[1]]
```

Out[45]=
$$\left\{C[1] \to -\frac{i\,e^{K\,\delta}\,\text{H1}\,\Omega}{-1 + e^{2\,K\,\delta}},\ C[2] \to \frac{i\,e^{K\,\delta}\,\text{H1}\,\Omega}{-1 + e^{2\,K\,\delta}}\right\}$$

Then

In[46]:=
```
w[16] = w[12] /. w[15] // RE
```

Out[46]=
$$\text{V1}[z][Z] == \frac{i\,e^{K\,\delta - K\,Z\,\delta}\,\text{H1}\,\Omega}{-1 + e^{2\,K\,\delta}} - \frac{i\,e^{K\,\delta + K\,Z\,\delta}\,\text{H1}\,\Omega}{-1 + e^{2\,K\,\delta}}$$

There remains the final boundary boundary condition

In[47]:=
```
w[11] [[4]]
```
Out[47]=
$$\mathbb{i}\, H1\, K^2 + \Omega\, V1[z]'[1] == 0$$

In[49]:=
```
w[17] = MapEqn[D[#, Z] &, w[16]] /. Z → 1 /. Sol[w[11] [[4]], V1[z]'[1]]  // Simplify
```
Out[49]=
$$-\frac{\mathbb{i}\, H1\, K \left(K - \frac{\left(1+e^{2 K \delta}\right) \delta\, \Omega^2}{-1+e^{2 K \delta}}\right)}{\Omega} == 0$$

or

In[50]:=
```
w[18] = MapEqn[(# Ω / (-I H1 K)) &, w[17]]
```
Out[50]=
$$K - \frac{\left(1 + e^{2 K \delta}\right) \delta\, \Omega^2}{-1 + e^{2 K \delta}} == 0$$

which constitutes the dispersion relation. This equation is usually expressed in terms of hyperbolic functions so some more manipulations are required

In[51]:=
```
w[19] = w[18] // ExpToTrig // Simplify
```
Out[51]=
$$K == \delta\, \Omega^2\, \mathrm{Coth}[K\, \delta]$$

In[53]:=
```
w[20] = MapEqn[(# / (Coth[K δ])) &, w[19]] // Reverse
```
Out[53]=
$$\delta\, \Omega^2 == K\, \mathrm{Tanh}[K\, \delta]$$

which is the classic result. Johnson actually considers the additional force associated with surface tension and derives a dispersion relation describing both classical water waves and capillary waves (the ripples one observes on water waves). Using the symbolic manipulation tools developed so far, it would be straightforward to incorporate this generalization. As a matter of good symbolic computing practice, it builds confidence to derive a a complicated result by starting with the simplest nontrivial case and then systematically adding physical effects.

2 Analysis of the dispersion relation

I analyze the dispersion relation derived in section 1.

In[54]:=
```
w2[1] = w[20]
```
Out[54]=
$$\delta\, \Omega^2 == K\, \mathrm{Tanh}[K\, \delta]$$

Return to dimensional parameters

In[55]:=
```
w2[2] = w2[1] /. {Ω → ω τ, K → k λ} /. δ → h/λ /. λ → v0 τ /. v0 → √(g h)
```

Out[55]=
$$\frac{h\,\tau\,\omega^2}{\sqrt{g\,h}} == \sqrt{g\,h}\;k\,\tau\,\text{Tanh}[h\,k]$$

Consider the deep (h → ∞) and shallow (h → 0) water limits.

In[56]:=
```
w2[3] = {w2[2] /. Tanh[_] → 1, w2[2] /. Tanh[x_] → x}
```

Out[56]=
$$\left\{\frac{h\,\tau\,\omega^2}{\sqrt{g\,h}} == \sqrt{g\,h}\;k\,\tau,\;\frac{h\,\tau\,\omega^2}{\sqrt{g\,h}} == h\,\sqrt{g\,h}\;k^2\,\tau\right\}$$

I begin consider the wavelength-period dispersion. Explicit expressions are available for the approximate forms

In[57]:=
```
w2[4] = w2[3] /. ω → 2 π / τ /. k → 2 π / λ
```

Out[57]=
$$\left\{\frac{4\,h\,\pi^2}{\sqrt{g\,h}\;\tau} == \frac{2\,\sqrt{g\,h}\;\pi\,\tau}{\lambda},\;\frac{4\,h\,\pi^2}{\sqrt{g\,h}\;\tau} == \frac{4\,h\,\sqrt{g\,h}\;\pi^2\,\tau}{\lambda^2}\right\}$$

In[58]:=
```
w2[5] = Solve[#, λ] & /@ w2[4]
```

Out[58]=
$$\left\{\left\{\left\{\lambda \to \frac{g\,\tau^2}{2\,\pi}\right\}\right\},\;\left\{\left\{\lambda \to -\sqrt{g}\;\sqrt{h}\;\tau\right\},\;\left\{\lambda \to \sqrt{g}\;\sqrt{h}\;\tau\right\}\right\}\right\}$$

The physically relevant solutions are

In[59]:=
```
w2[6] = {w2[5][[1, 1]], w2[5][[2, 2]]} // Flatten // RE
```

Out[59]=
$$\left\{\lambda == \frac{g\,\tau^2}{2\,\pi},\;\lambda == \sqrt{g}\;\sqrt{h}\;\tau\right\}$$

I define some functions to facilitate plots

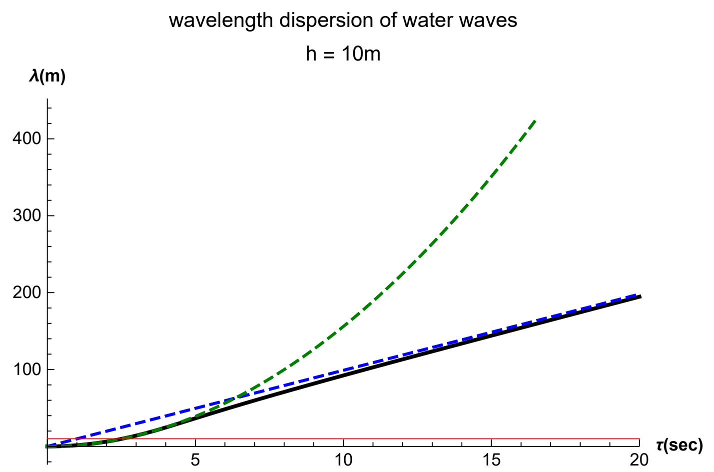
In[60]:=
```
Clear[WaterWaveLength, ShallowWaterApproximation, DeepWaterApproximation];
WaterWaveLength[τ_, g_, h_] := FindRoot[(2 π λ)/(g τ^2) == Tanh[(2 h π)/λ], {λ, 50}][[1, 2]];
ShallowWaterApproximation[τ_, g_, h_] := √g √h τ;
DeepWaterApproximation[τ_, g_, h_] := (g τ^2)/(2 π)
```

For a nominal water depth of h = 10 m, we see that waves with periods of less than 5 seconds (and correspondingly short wavelengths) are reasonably well described by the deep water approximation. For periods of order 10 seconds and wavelength of order 100m, the shallow water approximation is more accurate.

In[70]:=

```
Quiet@Module[{g = 9.8, h = 10, τMax = 20, refLine},
  refLine = {Red, Line[{{0, h}, {τMax, h}}]};
  Plot[{WaterWaveLength[τ, g, h], ShallowWaterApproximation[τ, g, h],
    DeepWaterApproximation[τ, g, h]}, {τ, 0, τMax},
   AxesLabel → {Stl["τ(sec)"], Stl["λ(m)"]}, Epilog → refLine, PlotStyle →
    {BLACK, Directive[Blue, Dashed], Directive[Darker[Green, 0.5], Dashed]},
   PlotLegends → {"general", "shallow", "deep"},
   PlotRange → {{0, τMax}, Automatic},
   PlotLabel → StringForm["wavelength dispersion of water waves\nh = ``m", h]]]
```

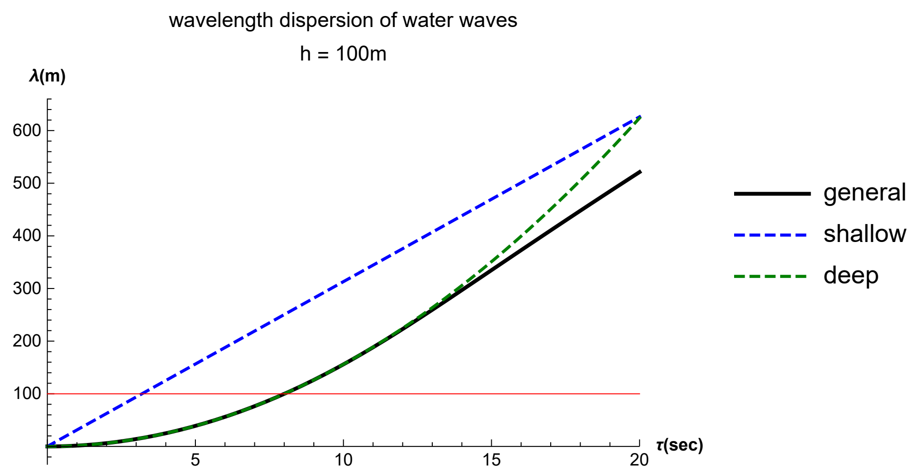
Out[70]=



For h = 100m, the deep water approximation is accurate for periods of almost 15 seconds.

In[71]:=
```
Quiet@Module[{g = 9.8, h = 100, τMax = 20, refLine},
  refLine = {Red, Line[{{0, h}, {τMax, h}}] };
  Plot[{WaterWaveLength[τ, g, h], ShallowWaterApproximation[τ, g, h],
    DeepWaterApproximation[τ, g, h]}, {τ, 0, τMax},
   AxesLabel → {Stl["τ(sec)"], Stl["λ(m)"]}, Epilog → refLine, PlotStyle →
    {BLACK, Directive[Blue, Dashed], Directive[Darker[Green, 0.5], Dashed]},
   PlotLegends → {"general", "shallow", "deep"},
   PlotRange → {{0, τMax}, Automatic},
   PlotLabel → StringForm["wavelength dispersion of water waves\nh = ``m", h]]]
```

Out[71]=



For reference, wind driven storm waves might have T ~ 10 sec, while a water wave driven by a seismic disruption might have T ~ 1000 sec.

I next consider the dispersion of the phase velocity

In[72]:=
```
w2[7] = Solve[w2[2] /. ω → v_ph k, v_ph][[2, 1]]
```

Out[72]= $v_{ph} \to \frac{\sqrt{g}\ \sqrt{\mathrm{Tanh}[h\,k]}}{\sqrt{k}}$

It is convenient to introduce a reference velocity $v_0 = \sqrt{g\,h}$ and a reference wavenumber κ = k h

In[76]:=
```
w2[8] =
 w2[7] /. Sol[v0 == √(g h), g] /. Sol[κ == k h, k] // Simplify[#, {v0 > 0, h > 0}] &
```

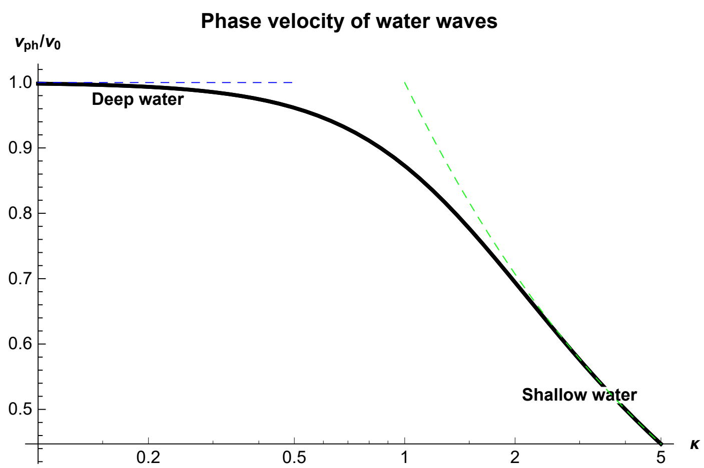
Out[76]= $v_{ph} \to \frac{v_0 \sqrt{\mathrm{Tanh}[\kappa]}}{\sqrt{\kappa}}$

Some brief comments are: Water waves are free of dispersion in deep water. In shallow water, waves “feel” the bottom and can disperse. The physical behavior of water waves is discussed in numerous texts and I will not repeat that discussion here. My primary focus has been to illustrate that symbolic manipulation can be used to carry out the derivation and subsequent analysis.

In[77]:=

```
Module[{v0 = 1, valsShallow, valsDeep},
 valsShallow =
  {Directive[{Green, Dashed}], Line@Table[{Log[κ], v0 / √κ}, {κ, 1, 5, .1}]};
 valsDeep = {Directive[{Blue, Dashed}],
   Line@Table[{Log[κ], v0}, {κ, 0.1, 0.5, .1}]};
 LogLinearPlot[{Labeled[Sqrt[Tanh[κ]/κ], Stl["Deep water"], {0.25, Below}],
   Labeled[Sqrt[Tanh[κ]/κ], Stl["Shallow water"], {3, Below}]},
  {κ, 0.1, 5}, AxesLabel → {Stl["κ"], Stl["vph/v0"]},
  PlotStyle → BLACK,
  PlotLabel → Stl["Phase velocity of water waves"],
  Epilog → {valsShallow, valsDeep}]]
```

Out[77]=

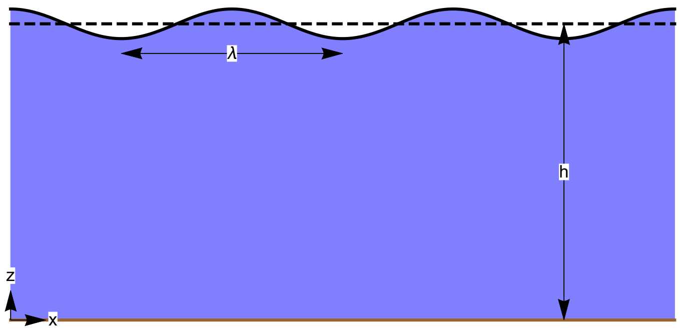


Appendix - Generation of figure

In[78]:=

```
Module[{ϵ = 0.05, h = 1, hLine, λLine, axes},
 axes = {Arrowheads[0.03], Arrow[{{0, 0}, {1, 0}}], Text["x", {1.2, 0}],
   Arrow[{{0, 0}, {0, 0.1}}], Text["z", {0, 0.15}]};
 hLine = {Directive[Black], Arrowheads[{-0.03, 0.03}],
   Arrow[{{5 π, 0}, {5 π, 1}}], Text["h", {5 π, 0.5}] };
 λLine = {Directive[Black], Arrowheads[{-0.03, 0.03}],
   Arrow[{{π, 0.9}, {3 π, 0.9}}], Text["λ", {2 π, 0.9}] };
 Plot[{1 + ϵ Cos[x], 0, h}, {x, 0, 6 π}, Filling → {1 → {2}},
  FillingStyle → Lighter[Blue, 0.5],
  PlotStyle → {Black, Brown, Directive[Black, Dashed]},
  Axes → False, AspectRatio → 1/2, Epilog → {axes, hLine, λLine}]]
```

Out[78]=



Functions

In[3]:=

```
Clear[DisplayEqns];
DisplayEqns[eqns_, subs_] :=
 Module[{neqns},
  neqns = Transpose[{Range[1, Length[eqns]], eqns}] /. subs // PhysicsForm]
```

In[5]:=

```
Clear[RefineEqn];
RefineEqn[lhs_ == rhs_, varList_] :=
 Module[{nEqn, nLhs, nRhs, cancellableCommonFactors},
   nEqn = StandardizeEqn[lhs == rhs];
  {nLhs, nRhs} = {nEqn[[1]], nEqn[[2]]};
  (*Print["{nLhs, nRhs} = ", {nLhs, nRhs}];*)
  cancellableCommonFactors =
   Select[Factor[nLhs], Or[NumericQ[#], ConstantQ[#, varList]] &];
  If[Or[cancellableCommonFactors == 0, cancellableCommonFactors == Factor[nLhs]],
   cancellableCommonFactors = 1];
  (*Print["numericalFactors} = ", cancellableCommonFactors];*)
  Simplify[nLhs / cancellableCommonFactors] ==
   Simplify[nRhs / cancellableCommonFactors]]
```