Hanbat National University

Course: Design of Coastal Engineering

Report II

November 29, 2022

Author	*Student ID*
Jonghyeok Kim	20201967

Report II

Jonghyeok Kim

November 29, 2022

Contents

1 Introduction

기본적으로 아래의 분산관계식(Dispersion Relationship)을 이용하여 본 보고서에 수록된 문제들을 해결할 수 있다.

$$\sigma^2 = gktanh(kh) \tag{1.1}$$

만약 kh의 값이 조건으로 제시가 되었다면, 위의 분산 관계식을 이용하여 σ^2으로부터 σ 값을 산정하고, 아래의 각진동수 σ에 대한 관계식으로부터,

$$\sigma = \frac{2\pi}{T} \tag{1.2}$$

다음과 같이 주기 T에 대한 식으로 전개하여 주기 T를 산정해낼 수 있다.

$$T = \frac{2\pi}{\sigma} \tag{1.3}$$

반면에, kh가 제시되어 있지 않거나, 파수 k의 값을 모르는 경우에는 위의 분산관계식에 대해 양해법(Explicit Method)과 음해법(Implicit Method)을 이용하여 k값을 산정할 수 있다.

본 보고서에서 필자는 음해법으로서 이분법(Bisection Method)을 이용하였고, Boundary Points를 결정함에 있어서 양해법 중 하나인 아래의 Eckart 식을 이용하였다.

$$\sigma^2 = gk\sqrt{tanh(\frac{\sigma^2}{g}h)} \tag{1.4}$$

반드시는 아니나, 분산관계식에서 파수 k를 구함에 있어 음해법(Implicit Method)을 사용하는 경우에는 양해법(Explicit Method)을 이용하여 구한 k값을 초깃값으로 하는 경우가 일반적이다.

또한, 위의 양해법에 대한 Eckart 식 외에도 다음과 같은 Hunt의 식도 존재한다.

$$(kh)^2 = y^2 + \frac{y}{1+\sum_{n=1}^{6} d_n y^n} \tag{1.5}$$

where,

$y = \frac{\sigma^2 h}{g}$

$d_1 = 0.6666666666$, $d_2 = 0.3555555555$, $d_3 = 0.1608465608$, $d_4 = 0.0632098765$, $d_5 = 0.0217540484$, $d_6 = 0.0076507983$

특히, You(2003)에 따르면, 여러 개의 파랑분산식의 양해(Hunt, 1979; Nielsen, 1982; Fenton과 McKee, 1990; Guo, 2002; Nielsen, 2002; You, 2003)를 비교하였을 때, Hunt의 해가 $\nu \leq \pi$의 수심범위에서 가장 정확하다는 것을 밝혀 내었다. 이러한 사실로부터, 초깃값 산정의 방법에 있어서 Eckart 식 외에 Hunt의 식 사용을 고려해보는 것도 음해법을 이용한 해의 수렴속도를 증가시키는 한 가지 요인으로 작용할 수 있을 것으로 기대해볼 수 있겠다.

물론 양해법에는 Eckart의 방법과 Hunt의 방법 외에도 여러 개의 다른 양해법이 존재한다.

본 과제에서는 위의 기본적인 해안공학에서의 개념에 기반하여, 규칙파의 변형 중 천수변형에 대해서 살펴볼 예정이다.

파랑의 천수변형은 반사와 파 동력의 소산이 없다는 가정하에서 유도될 수 있다.

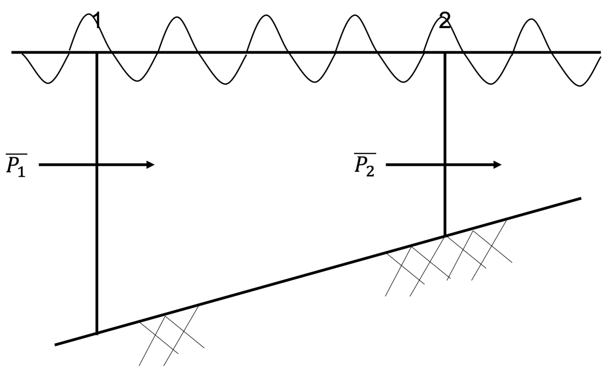


Figure 1: Wave Shoaling

즉, 위의 Figure 1 상에서는 수심의 변화가 있어 실제 상황에서는 파 반사가 존재하지만, 이것이 발생하지 않는다는 가정과 1번 지점과 2번 지점에서의 파 동력의 소산이 없다는 가정 하에서 아래의 식이 성립한다.

$$P_1 = P_2 \tag{1.6}$$

또한, $P = ECn$이므로 위의 식 (1.6)을 아래와 같이 다시 작성할 수 있다.

$$(ECn)_1 = (ECn)_2 \tag{1.7}$$

추가적으로, $E = \frac{1}{8}\rho g H^2$이므로, 위의 식 (1.7)을 다시 한번 아래와 같이 작성할 수 있다.

$$\frac{1}{8}\rho g H_1^2 C_1 n_1 = \frac{1}{8}\rho g H_2^2 C_2 n_2 \tag{1.8}$$

위의 식 (1.8)을 기지의 파고 H_1에 대한 미지의 파고 H_2에 대한 식으로 변형하면 다음과 같고, 이는 천수계수 K_s로 정의된다.

$$K_s = \frac{H_2}{H_1} = \sqrt{\frac{C_1 n_1}{C_2 n_2}} = \sqrt{\frac{L_1 n_1}{L_2 n_2}} \tag{1.9}$$

여기서 무차원 계수 n은 다음과 같이 정의된다.

$$n = \frac{1}{2}\left(1 + \frac{2kh}{sinh(2kh)}\right) \tag{1.10}$$

특히, 심해(Deep Water)에서는 $kh \geq \pi$ 즉, $kh \to \infty$이므로, 무차원 계수 n을 구성하고 있는 $\frac{2kh}{sinh(2kh)} \to 0$가 되어, $n \approx 0.5$이다. 그러므로 군속도 $C_g = nC_0 = 0.5C_0$로 그 값이 일정(Constant)하게 된다.

이로부터, 위의 식 (1.9)에서 만약 심해에서 파가 입사되어 올 때의 파고를 계산해야하는 상황이라면, 심해를 기준지점으로 설정하여, 다음과 같이 n값을 0.5로 설정하여 임의지점의 파고를 계산할 수 있다.

$$K_s = \frac{H_2}{H_1} = \sqrt{\frac{C_1 \cdot 0.5}{C_2 \cdot n_2}} \tag{1.11}$$

2 Questions

수심 $h_0 = 10m$인 상황에서 파고 $H = 2.0m$인 파가 입사하고 있다.

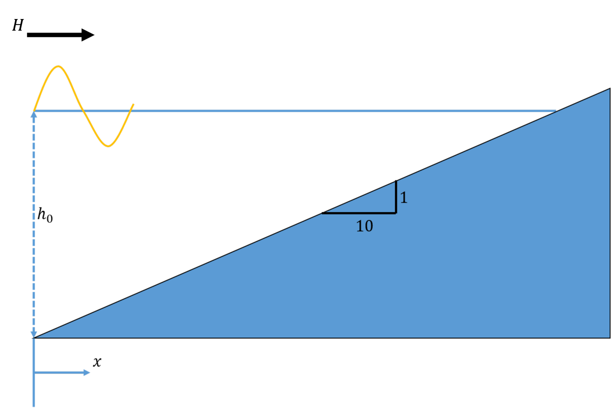


Figure 2: Condition #1

2.1 Question #1

수심 h_0에서 $kh = \pi$를 만족하는 주기 T_0를 구하시오.

2.2 Answer #1

먼저 먼 바다에서 파가 진행하면 해당 파에 대해서는 항상 주기가 일정하므로, 파고 계산시 변수 1개를 줄일 수 있다. 이러한 주기의 산정을 위해 Questions 섹션에서 주어진 수심 $h_0 = 10m$와 Question #1 섹션에서 조건으로 주어진 $kh = \pi$를 이용하여 k값을 아래와 같이 산정한다.

$$kh = k \times 10m = \pi \tag{2.1}$$

$$\therefore k = \tfrac{\pi}{10m} \tag{2.2}$$

위에서 도출한 k값으로부터, 분산관계식 (1.1)을 이용하여 다음과 같이 각진동수 σ^2을 산정한다.

$$\sigma^2 = 9.81m/sec^2 \times \tfrac{\pi}{10m} \times tanh(\tfrac{\pi}{10m} \times 10m) \tag{2.3}$$

$$\therefore \sigma^2 = 3.0704133(rad^2/sec^2) \tag{2.4}$$

위의 식 (2.3)로부터 도출한 σ^2의 값으로부터, 각진동수 σ와 주기 T의 관계식 (1.2)에서 주기 T를 산정하기 위해 (2.4)의 값에 제곱근을 취하여 σ의 값을 다음과 같이 산정한다.

$$\sigma = \sqrt{\sigma^2} = \sqrt{3.0704133(rad^2/sec^2)} \approx 1.7522594830545946(rad/sec) \tag{2.5}$$

위의 (2.5)에서 구한 σ 값을 이용하여 각진동수와 주기 T의 관계식 (1.3)을 통해 주기 T를 다음과 같이 산정한다.

$$T = T_0 = \tfrac{2\pi}{\sigma} = \tfrac{2\pi}{1.7522594830545946} \approx 3.58(sec) \tag{2.6}$$

2.3 Question #2

시작점에서 $\Delta x = 1m$ 간격으로 파고의 높이를 계산하여 그래프를 그리시오.

2.4 Answer #2

먼저 (1.11) 식의 C_1을 앞서 Question #1 섹션에서 산정한 파수 k와 각진동수 σ를 다음의 파속 C에 대한 식에 대입하여 C_1을 산정한다.

$$C = C_1 = \frac{L}{T} = \frac{\frac{2\pi}{k}}{\frac{2\pi}{\sigma}} = \frac{\sigma}{k} \tag{2.7}$$

$$C_1 = \frac{1.7522594830545946}{0.3141592653589793} \tag{2.8}$$

$$\therefore C_1 = 5.577615166155759(m/sec) \tag{2.9}$$

그 다음, 수심 $h_0 = 10m$인 상황에서 파고 $H = 2.0m$인 파가 입사하고 있고, 그때의 파는 $kh = \pi$를 만족하고, 이는 $kh \geq \pi$로 심해영역이므로($n = 0.5$) 앞서 Introduction 섹션에서 살펴본 파고를 계산하는 식 (1.11)을 이용하여 임의지점에서의 파고를 계산할 수 있다.

시작점인 수심 $h_0 = 10m$를 기준으로 하여, Figure 2에 도시한 x축의 양의 방향으로 $\Delta x = 0.1m$ 간격으로 잘라낸다.

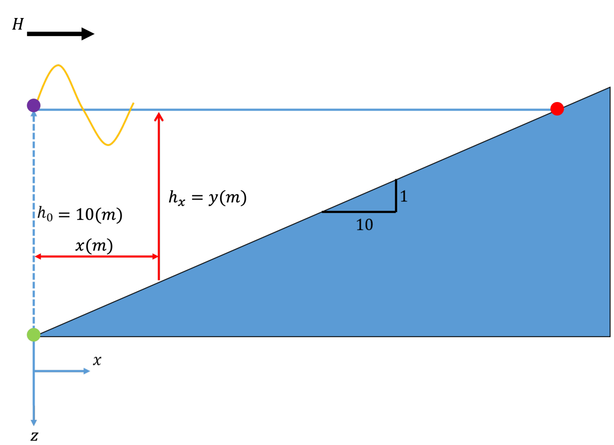


Figure 3: Condition #2

다음으로, Figure 3에서 도시한 보라색 점의 좌표를 $(0,0)$으로 설정하였을 때, 연두색 점의 좌표는 $(0,10)$이 되고, 기울기는 $-\frac{1}{10}$이므로, 이로부터 직선의 방정식을 다음과 같이 유도하여 임의로 원점에서 x만큼 이동하였을 때, 원점에서 해당 지점까지 이동한 거리에서의 수심 $y = h_x$를 산정한다.

$$y = h_x = -\frac{1}{10}(x - 0) + 10 \tag{2.10}$$

유도한 직선의 방정식 (2.10)에서 $y = h_x = 0m$일 때, 즉 수심이 0m일 때의 원점에서 이동한 거리 x를 계산하면 다음과 같다.

$$y = 0 = -\frac{1}{10}(x) + 10 \tag{2.11}$$

$$\therefore x = 100m \tag{2.12}$$

그러므로, $x = 0m$에서 $x = 99m$까지 $x_0, x_1, x_2, \cdots, x_{99}$ 각각에 대해 총 100개의 수심 $h_0, h_1, h_2, \cdots, h_{99}$을 산정한다.

산정한 각 수심에 대해 음해법(Implicit Method)인 이분법(Bisection Method)을 이용하여 미지의 파수 k값을 산정하기 위해, 그 초깃값을 양해법(Explicit Method) 중 하나인 Eckart 식 (1.4)을 이용하여 산정한다.

예를 들어, $x_1 = 1m$에서 수심 h_1의 값은 $9.9m$이고, 먼 바다에서 진행되어 온 동일 파에 대해서는 주기 T가 동일하므로, 앞서 Question #1 섹션에서 산정한 T_0와 그 값이 동일하다. 즉, $T = T_0$이다. 또한, Eckart 식의 σ^2 값 역시 동일 주기이므로, (1.2) 관계식으로부터 이것 역시 앞서 Question #1 섹션에서 산정한 각진동수 σ의 제곱한 값 $\sigma^2 = 3.0704133(rad^2/sec^2)$과 동일하다. 마지막으로, 중력 가속도 $g = 9.81m/sec^2$이라 하면, Eckart 식 (1.4)을 k에 대한 식으로 정리하여, 위의 값들을 대입하여 초기 k값을 산정해낼 수 있다.

$$k = \frac{\sigma^2}{g\sqrt{tanh(\frac{\sigma^2}{g}h)}} = \frac{3.0704133}{9.81\sqrt{tanh(\frac{3.0704133}{9.81}9.9)}} \tag{2.13}$$

$$\therefore k = 0.31362574(rad/m) \tag{2.14}$$

그때의 k의 초깃값은 $0.31362574(rad/m)$로 도출된다.

산정한 초깃값을 이용하여, 음해법(Implicit Method) 중 하나인 이분법(Bisection Method)을 통해 컴퓨터를 이용하여 수치해를 구해보자.

이를 위해, 우선 분산관계식(Dispersion Relationship) (1.1)의 좌변이 0이 되도록 모든 항을 이항하여 정리하면 다음과 같다.

$$0 = \frac{gktanh(kh)}{\sigma^2} - 1 \tag{2.15}$$

위의 식 (2.15)에 대해 오차(Error)를 0.5×10^{-6}으로 설정하여, 위에서 산정한 (2.13)의 초기 k값의 ± 0.1만큼을 이분법의 해 탐색 구간으로 설정하여 k 값을 산정한다.

예를 들어, 위에서와 같이 수심 $h = 9.9m$에 대해 이분법(Bisection Method)을 이용하여 k값을 산정하면 그 값은 $k = 0.31423356(rad/m)$으로, 초깃값과의 백분율 차이는 0.06% 수준이다.

다음의 파속 C 식에 수치해석 기법인 이분법(Bisection Method)으로 도출한 k값과 Question #1에서 산정한 σ 값을 대입하여 (1.11)의 파속 C_2를 산정한다.

$$C = C_2 = \frac{L}{T} = \frac{\frac{2\pi}{k}}{\frac{2\pi}{\sigma}} = \frac{\sigma}{k} \tag{2.16}$$

$$C_2 = \frac{1.7522594830545946}{0.31423356} \tag{2.17}$$

$$\therefore C_2 = 5.57629636(m/sec) \tag{2.18}$$

또한, Introduction 섹션에서 살펴본 무차원 계수 n에 대한 식 (1.10)에도 마찬가지로 $k = 0.31423356$와 $h = 9.9$를 대입하여 식 (1.11)의 n_2를 산정한다.

$$n = n_2 = \frac{1}{2}\left(1 + \frac{2 \cdot 0.31423356 \cdot 9.9}{sinh(2 \cdot 0.31423356 \cdot 9.9)}\right) \tag{2.19}$$

$$\therefore n_2 = 0.51235422 \tag{2.20}$$

마지막으로, 식 (1.11)을 임의지점에서의 미지의 파고 H_2에 대한 식으로 다음과 같이 정리한다.

$$H_2 = H_1 \cdot \sqrt{\frac{C_1 \cdot 0.5}{C_2 \cdot n_2}} \tag{2.21}$$

그 다음, 기준지점에서 주어진 파고 $H_1 = 2.0m$와 위에서 산정한 C_1, C_2, 그리고 n_2의 값을 다음과 같이 대입하여 수심 $h_1 = 9.9m$일 때의 파고 H_2 값을 산정해낸다.

$$H_2 = 2.0 \cdot \sqrt{\frac{5.577615166155759 \cdot 0.5}{5.57629636 \cdot 0.51235422}} \tag{2.22}$$

$$\therefore H_2 = 1.9759738287692346m \tag{2.23}$$

위의 원리를 그대로 적용하여 나머지 수심 $h_2, h_3, h_4, \cdots, h_{99}$에서의 파고 H를 구하여 이를 수심(Depth) h에 대한 파고(Height of a Wave) H 그래프로 나타내면 다음과 같다.

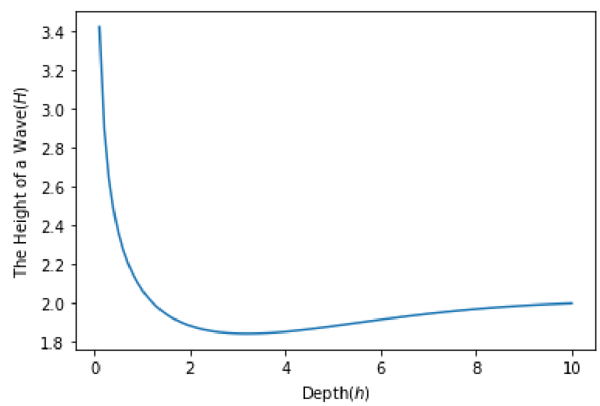


Figure 4: Graph of Wave Height(H) vs. Depth(h) When $n_1 = 0.5$

파고 H를 원점으로부터 떨어진 거리(Distance) x에 대한 그래프로 나타내면 다음과 같다.

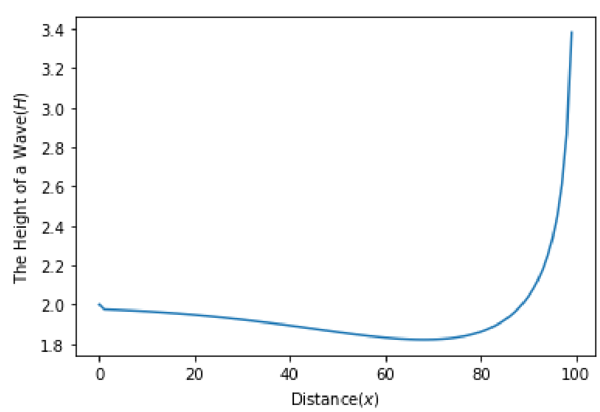


Figure 5: Graph of Wave Height(H) vs. Distance(x) When $n_1 = 0.5$

위의 거리에 대한 파고 그래프 Figure 5를 확인해보면, 0m 부근에서 불연속점으로 인해 매끄러운 곡선(Smooth Curve)가 도출되지 못하고 있음을 확인할 수 있다. 이는 앞서 심해에서의 무차원 계수 $n = n_1$을 별도로 무차원

계수 식 (1.10)에 의해 산정하지 않고, 다음 사항에 의거하여,

« 특히, 심해(Deep Water)에서는 $kh \geq \pi$ 즉, $kh \rightarrow \infty$이므로, 무차원 계수 n을 구성하고 있는 $\frac{2kh}{sinh(2kh)} \rightarrow 0$가 되어, $n \approx 0.5$이다. 그러므로 군속도 $C_g = nC_0 = 0.5C_0$로 그 값이 일정(Constant)하게 된다.»

즉, kh값이 π보다 커짐에 따라 무차원 계수 n의 값이 0.5에 가까워질 것이므로, 단순히 0.5의 값을 n_1의 값으로 택함에 있다.

이로부터 앞서 n_1 역시 n_2와 동일한 방식으로 무차원 계수 n의 식 (1.10)에 값을 대입하여 n_1을 산정해보면 그 값은 $n_1 = 0.51173353$이다.

n_1의 값으로부터 파고 H를 다시 계산하고, 이를 수심(Depth) h와 원점으로부터 떨어진 거리(Distance) x에 대해 다시 그래프를 그려보면 다음과 같다.

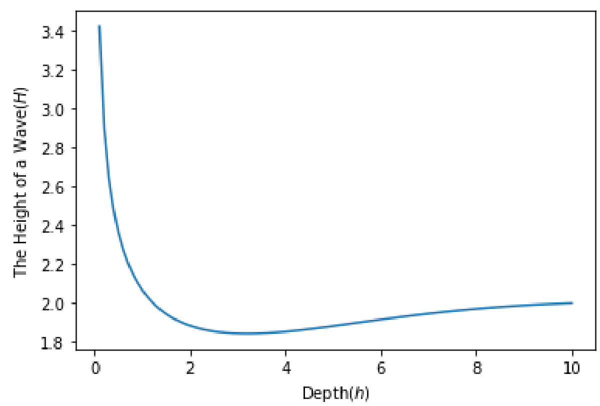


Figure 6: Graph of Wave Height(H) vs. Depth(h) When $n_1 = 0.51173353$

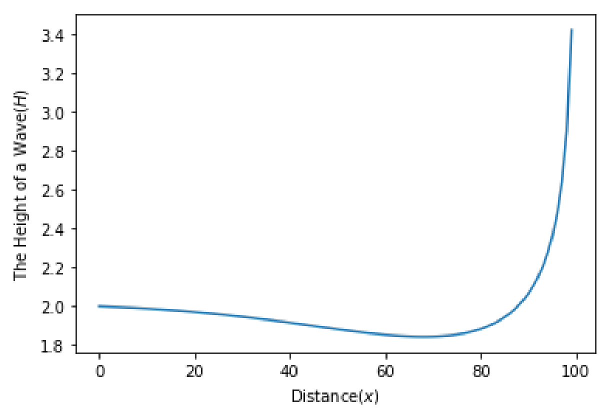


Figure 7: Graph of Wave Height(H) vs. Distance(x) When $n_1 = 0.51173353$

이전과는 다르게, Figure 7의 $0m$ 부근에서 연속적으로 매끄러운 곡선(Smooth Curve)이 도출됨을 확인할 수

있다.

이번에는 양해법(Explicit Method)인 Eckart와 Hunt의 방법으로부터 파수 k를 산정하고, 이들의 그래프 비교 및 차이에 대한 평균 백분율을 분석한 다음, 양해법(Explicit Method)인 Eckart의 방법과 Hunt의 방법에서 구한 파수 k로부터 파고 H와 음해법(Implicit Method)인 이분법(Bisection Method)으로 도출한 파수 k로부터 구한 파고 H를 서로 비교하고자 한다.

먼저 파수 k를 수심 $h_0 = 10m$일 때의 파수를 포함하여, 수심 $h_1, h_2, h_3, \cdots, h_{99}$까지에 대해 Eckart 식 (1.4)과 Hunt 식 (1.5)으로부터 구한 다음, 위에서 이분법에 의해 도출한 파수 k를 수심 h에 대해 그래프로 비교해보면 다음과 같다. (이때의 무차원 계수 n은 심해에서의 $n = 0.5$값을 이용하지 않고, (1.10)식으로부터 산정한 값을 이용하였다.)

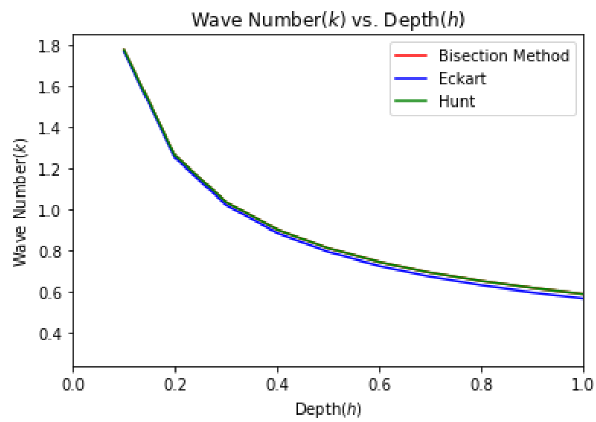


Figure 8: Graph of Wave Number(k) vs. Depth(h)

Figure 8에서 Bisection Method와 Hunt's Method의 파수 k는 큰 차이를 보이고 있지 않지만, Eckart's Method의 파수 k는 상대적으로 큰 차이를 보이면서 나머지 두 방법에 의해 산정한 수심 h에 대한 파수 k와는 다르게 불연속적으로 직선의 개형을 보임도 확인할 수 있다.

Bisection Method로 산정한 파수 k에 대해 나머지 두 방법에 의해 산정한 파수 k와의 차이의 평균의 백분율 값을 비교하여 표로 정리하면 다음과 같다.

Table 2.3.1 Bisection Method vs. Eckart Method vs. Hunt's Method for k

	k (Using Eckart's Method)	k (Using Hunt's Method)
k (Using Bisection Method)	1.00575799%	0.02306469%

Bisection Method로 산정한 파수 k를 기준으로 이 값과의 차이의 평균의 백분율 값으로서 Eckart's Method에서는 약 1.00%로 약 0.023%인 Hunt's Method에 비해 상대적으로 큰 차이를 보이고 있음을 확인해볼 수 있다.

또한, 각 방법에서 도출해낸 파수 k 값으로부터 식 (2.16)을 이용하여 파속 C를 산정하여 이를 수심 h에 대해 그래프를 작도하여 서로 비교해보면 다음과 같다.

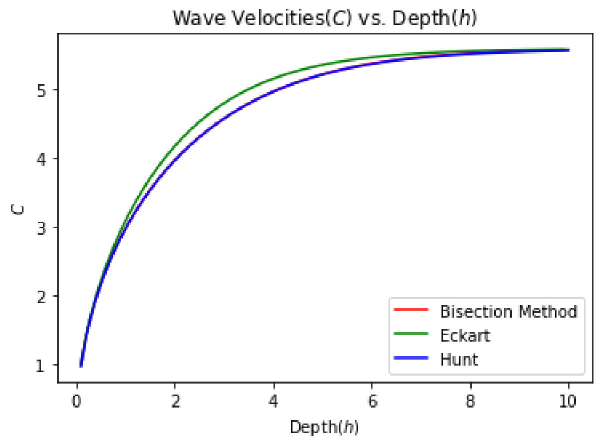


Figure 9: Graph of Wave Speed(C) vs. Depth(h)

각 방법에 대해 파수 k를 수심 h에 대해 작도한 Figure 8와 마찬가지로 Figure 9에서 파속 C 역시 Bisection Method와 Hunt's Method에서는 큰 차이가 발생하지 않음을 육안으로 확인할 수 있으나, Bisection Method와 Eckart's Method에서는 육안으로 확인할 수 있을 정도의 차이가 발생하고 있음을 확인해볼 수 있다.

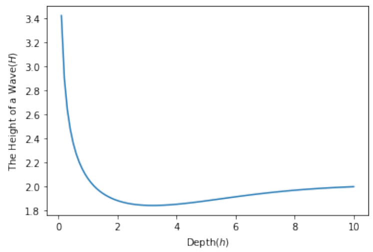
마찬가지로, Bisection Method로 산정한 파속 C에 대해 나머지 두 방법에 의해 산정한 파속 C와의 차이의 평균의 백분율 값을 비교하여 표로 정리하면 다음과 같다.

Table 2.3.2 Bisection Method vs. Eckart Method vs. Hunt's Method for C

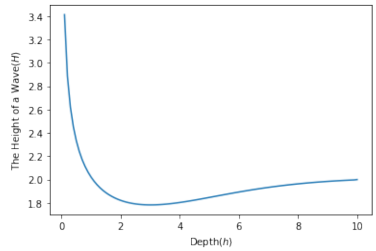
	C (Using Eckart's Method)	C (Using Hunt's Method)
C (Using Bisection Method)	10.77090314%	0.39054112%

Bisection Method로 산정한 파속 C값을 기준으로 이 값과의 차이의 평균의 백분율 값으로서 Eckart's Method에서는 약 10.77%로 상당히 큰 차이를 보여주고 있음을 확인할 수 있고, Hunt's Method에서는 0.39%로 Eckart's Method에 비해서는 상대적으로 작은 차이를 보임도 확인해볼 수 있다.

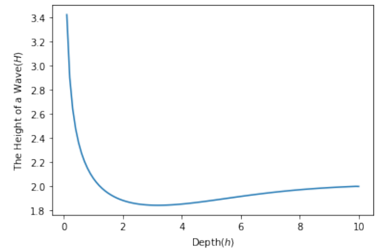
마지막으로, 위에서 Bisection Method를 통해 작도한 수심 h에 대한 파고 H의 그래프와 나머지 두 방법을 통해 작도한 수심 h에 대한 파고 H의 그래프를 비교해보자. (물론 이때의 기준이 되는 수심 h_0에서의 무차원 계수 n값은 알려진 심해에서의 $n = 0.5$의 값을 사용하지 않고, 식 (1.10)을 이용하여 정확한 값을 산정하여 이를 사용하였다.)



(a) Graph of Wave Height(H) vs. Depth(h) (Using Bisection Method)



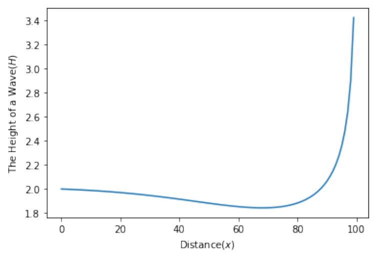
(b) Graph of Wave Height(H) vs. Depth(h) (Using Eckart's Method)



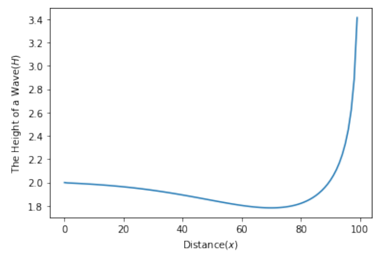
(c) Graph of Wave Height(H) vs. Depth(h) (Using Hunt's Method)

Figure 10: Graphs of Wave Height(H) vs. Depth(h)

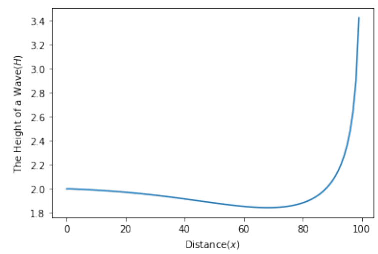
다음으로, 원점으로부터 이동한 거리(Distance) x에 대해 파고 H를 비교해보면 다음과 같다.



(a) Graph of Wave Height(H) vs. Distance(x) (Using Bisection Method)



(b) Graph of Wave Height(H) vs. Distance(x) (Using Eckart's Method)



(c) Graph of Wave Height(H) vs. Distance(x) (Using Hunt's Method)

Figure 11: Graphs of Wave Height(H) vs. Depth(h)

위의 Figure 10와 Figure 11의 모든 그래프 상에서 개형상에서의 변화를 보이고 있지는 않지만, Bisection Method로부터 산정한 파고 H 값을 기준으로 하여, Eckart's Method와 Hunt's Method로 산정한 파고 H 값의 차이의 평균값의 백분율을 표로 나타내면 다음과 같다.

Table 2.3.3 Bisection Method vs. Eckart Method vs. Hunt's Method for H

	H (Using Eckart's Method)	H (Using Hunt's Method)
H (Using Bisection Method)	4.498388630255303%	0.08965452942100072%

Bisection Method로 산정한 파고 H의 값을 기준으로 하였을 때, Eckart's Method로 산정한 파고 H 값과의 차이의 평균값의 백분율은 대략 4.5% 정도의 차이가 발생함을 확인하였고, Hunt's Method로 산정한 파고 H 값과의 차이의 평균값의 백분율은 대략 0.09% 정도의 차이가 발생함을 확인하였다.

3 Conclusion

3.1 Conclusion

본 보고서에서는 서론에서 언급하였듯이, 기본적으로 아래와 같이 정의되는 분산관계식(Dispersion Relationship)을 이용하여 모든 문제를 해결하였다.

$$\sigma^2 = gktanh(kh)$$

첫 과제로서, kh가 주어지는 경우에는 단순히 위의 분산관계식에 그 값을 대입 또는 주어진 수심 h로부터 k값을 도출하여 이를 대입함으로써, 각진동수 σ를 구하였고, $T = \frac{2\pi}{\sigma}$의 관계식으로부터 주기 T를 산정할 수 있었다.

또한, 두 번째 이후의 과제에서는 첫 번째 과제에서와는 다르게, k값이 주어지지 않은 상황에서 k 값을 산정하고 이로부터 파속 C를 구하는 과정이 수반되었다.

특히, Question #2에서와 같이 임의수심에서의 파고를 계산하기 위하여 미지인 k를 산정해야하는 경우에는 Eckart나 Hunt 식과 같은 양해법(Explicit Method)을 이용하여 k값 산정 및 음해법(Implicit)에 대한 초깃값으로 이를 이용하여, 더 정확하고 정밀한 해를 찾아보는 과정을 살펴보았다. 또한, 파수 k를 산정한 이후에는 파속 C에 대한 식 (2.16)을 이용하여 이것을 구하는 절차를 밟았다.

그 다음, 주어진 파고 H_1, 그리고 산정한 파속 C와 식 (1.10)을 이용하여 임의수심에서의 무차원계수 n을 산정하고, 식 (1.11)을 통해 임의수심에서의 파고 H_2를 계산할 수 있었다.

물론, 식 (1.11)의 경우에는 n_1의 값이 0.5인데, 이는 심해(Deep Water)에서는 $kh \geq \pi$ 즉, $kh \to \infty$이므로, 무차원 계수 n을 구성하고 있는 $\frac{2kh}{sinh(2kh)} \to 0$가 되어, $n \approx 0.5$가 되기 때문이다. 즉 이러한 심해(Deep Water)의 조건을 사용할 수 있는 이유는 Question #1 섹션에서 수심 h_0에서의 $kh = \pi$라는 조건에서 $kh \geq \pi$이므로 기준이 되는 수심 h_0에서 심해(Deep Water)가 되기 때문이다.

또한, 입사하는 하나의 파에 대해서는 그 주기(Periods) T가 일정하다는 조건을 이용할 수 있기 때문에 $kh = \pi$라는 조건과 분산관계식(Dispersion Relationship) (1.1)을 이용하여 σ^2의 값을 산정하고, 이로부터 식 (1.3)을 통해 주기 T를 도출하여 임의지점에서의 파고 H_2를 산정함에 있어 이를 고정값으로 사용할 수 있었다.

추가적으로, 음해법(Implicit Method)인 이분법(Bisection Method)으로 파수 k와 파속 C, 파고 H를 양해법(Explicit Method)인 Eckart의 방법(Eckart's Method)과 Hunt의 방법(Hunt's Method)과 비교하였고, 그 결과는 다음과 같았다.

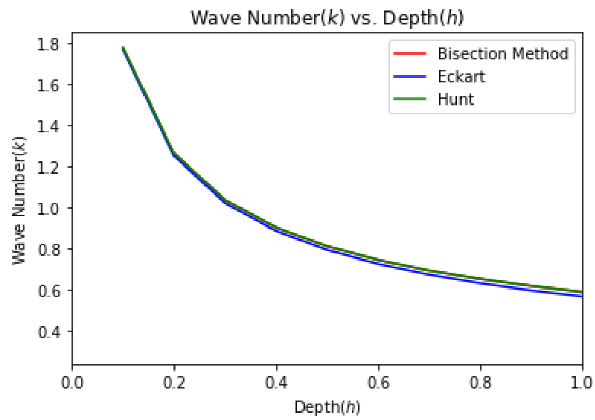


Figure 12: Graph of Wave Number(k) vs. Depth(h)

Table 2.3.1 Bisection Method vs. Eckart Method vs. Hunt's Method for k

	k (Using Eckart's Method)	k (Using Hunt's Method)
k (Using Bisection Method)	1.00575799%	0.02306469%

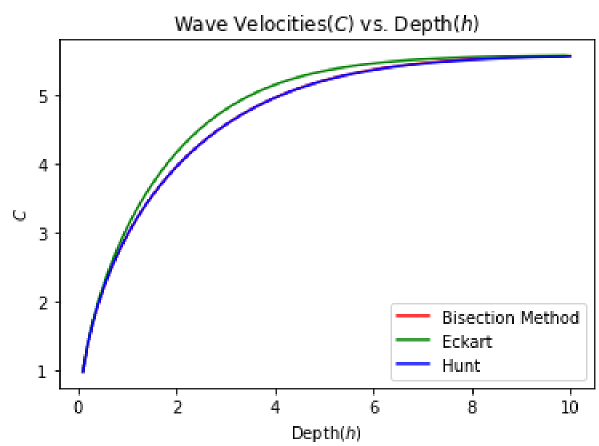
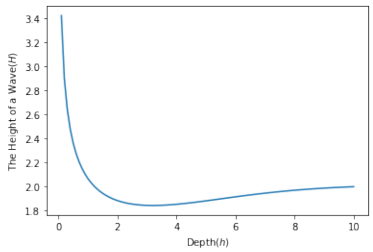


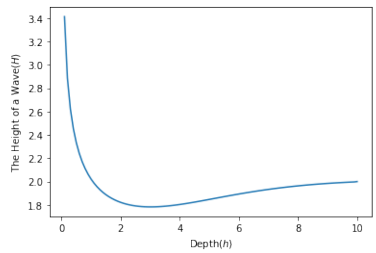
Figure 13: Graph of Wave Speed(C) vs. Depth(h)

Table 2.3.2 Bisection Method vs. Eckart Method vs. Hunt's Method for C

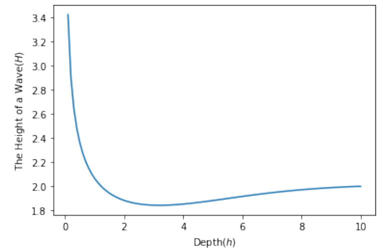
	C (Using Eckart's Method)	C (Using Hunt's Method)
C (Using Bisection Method)	10.77090314%	0.39054112%



(a) Graph of Wave Height(H) vs. Depth(h) (Using Bisection Method)

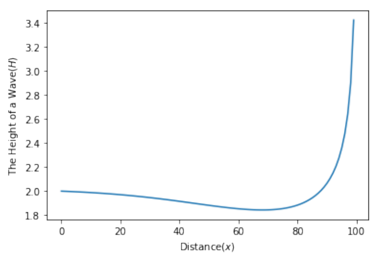


(b) Graph of Wave Height(H) vs. Depth(h) (Using Eckart's Method)

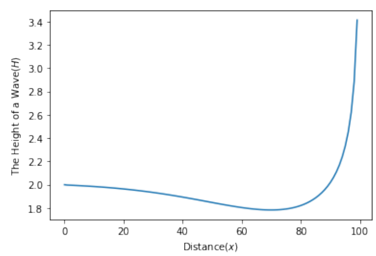


(c) Graph of Wave Height(H) vs. Depth(h) (Using Hunt's Method)

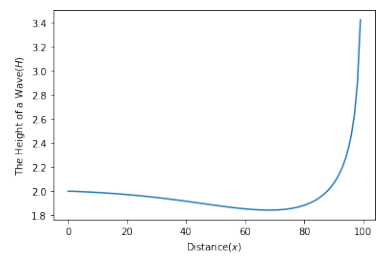
Figure 14: Graphs of Wave Height(H) vs. Depth(h)



(a) Graph of Wave Height(H) vs. Distance(x) (Using Bisection Method)



(b) Graph of Wave Height(H) vs. Distance(x) (Using Eckart's Method)



(c) Graph of Wave Height(H) vs. Distance(x) (Using Hunt's Method)

Figure 15: Graphs of Wave Height(H) vs. Depth(h)

Table 2.3.3 Bisection Method vs. Eckart Method vs. Hunt's Method for H

	H (Using Eckart's Method)	H (Using Hunt's Method)
H (Using Bisection Method)	4.498388630255303%	0.08965452942100072%

위의 결과로부터, Eckart의 방법은 Bisection Method보다 상대적으로 그 차이가 크게 발생하였으나, Hunt의 방법은 Eckart의 방법에 비해 상대적으로 작은 차이를 보임을 확인할 수 있었다.

이러한 결과는, You(2003)에 따르면, 여러 개의 파랑분산식의 양해(Hunt, 1979; Nielsen, 1982; Fenton과 McKee, 1990; Guo, 2002; Nielsen, 2002; You, 2003)를 비교하였을 때, Hunt의 해가 $\nu \leq \pi$의 수심범위에서 가장 정확하다는 것을 밝혀 내었는데, 수심 h에 대한 $y = \nu = \frac{\sigma^2 h}{g}$의 그래프를 도출하고, 그때의 최댓값 역시 도시해보면 다음과 같고,

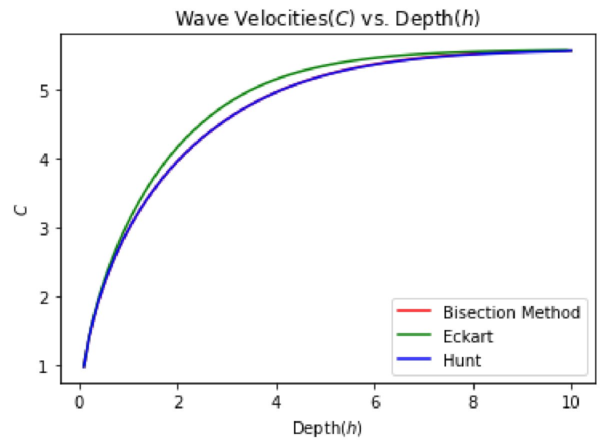


Figure 16: Graph of $y = \nu = \frac{\sigma^2 h}{g}$ vs. Depth(h)

그때의 최댓값이 3.12988104으로 이는 π보다 작기 때문에 즉, $\nu \leq \pi$ 범위에 존재하기 때문에 양해법(Explicit Method) 중에서 가장 정확한 해가 도출되었다는 결론을 내릴 수 있다.

즉, 위의 연구결과가 위에서 Hunt's Method가 Bisection Method와 크게 차이가 발생하지 않다는 사실을 뒷받침하고 있음도 확인할 수 있다.

결론적으로, 본 보고서에서는 다양한 방법을 이용하여 분산관계식을 통해 파수 k를 산정한 이후에는 파속 C를 산정하고, 무차원계수 n을 산정하여 최종적으로는 파고 H을 산정하는 과정이 수록되었다. 또한, 기본적인 해안공학의 개념들을 컴퓨터(Computers)내에서 코딩(Coding)을 통해 직접 구현해보며 이론적인 내용들을 재정립함과 동시에 실무적 활용 능력 또한 배양할 수 있었다.

4 Acknowledgement

본 과제를 부여해주신 Prof. Jung, T.H. 교수님께 감사의 말씀을 드립니다.

5 References

[1] Jonghyeok Kim. *Code Files Written in Python by Kim, J.H. For Report II in Design of Coastal Engineering*. 2022. URL: `https://github.com/enfycius/Coastal-Engineering/blob/main/Assign-2.ipynb` (visited on 11/10/2022).

[2] 장호철 이창훈. "순환 관계에 의한 파랑분산식의 양해". In: *대한토목학회 논문집* (2008), pp. 111–114.

Appendix A

Code Snippets

A Question #1

```
import math
import numpy as np
```

```
h0 = 10
H1 = 2.0
g = 9.81
```

```
h = h0
k = math.pi / h
```

```
k0 = k
```

```
k
```

```
sqr_sigma = g * k * np.tanh([k*h])
```

```
sqr_sigma
```

```
sigma = math.sqrt(sqr_sigma)
```

```
sigma
```

```
T0 = 2 * math.pi / sigma
```

```
T0
```

```
sigma / k0
```

B Question #2

```
C1 = sigma / k0
```

```
C1
```

```
N1 = 0.5
```

```
N1
```

```
h = []
k = []
C = []
N = []
H = []
```

```
def f(x, sqr_omega, h1):
    return g * (x / sqr_omega) * np.tanh([x*h1]) - 1
```

```
def bisection_method(k_init, sqr_omega, h1):
    error=0.5 * 10**(-6)

    a = -0.01 + k_init
    b = 0.01 + k_init

    while (b - a) / 2 > error:
        c = (b + a) / 2
        if f(c, sqr_omega, h1) == 0:
            break
        elif f(a, sqr_omega, h1)*f(c, sqr_omega, h1) > 0:
            a = c
        else:
            b = c
    c = (b + a) / 2

    return c
```

```
H.append(H1)
```

```
eckart_k = []
```

```
h.append(h0)
```

```
for i in range(0, 99):
    h1 = round(10 - (1/10)*(i+1), 2)
    k_init = sqr_sigma / (g * math.sqrt(np.tanh([sqr_sigma / g * h1])))
    k1 = abs(bisection_method(k_init, sqr_sigma, h1))
    C2 = sigma / k1
    N2 = 1/2 * (1 + 2*k1*h1/np.sinh([2*k1*h1]))
    H2 = H1 * math.sqrt((C1 * N1) / (C2 * N2))
    print(h1)
```

```
    eckart_k.append(k_init)
    h.append(h1)
    k.append(k1)
    C.append(C2)
    N.append(N2)
    H.append(H2)
```

```
k[0]
```

```
eckart_k[0]
```

```
abs(eckart_k[0] - k[0]) * 100
```

```
len(H)
```

```
import matplotlib.pyplot as plt
```

```
X = np.arange(0, 100)
```

```
X
```

```
plt.plot(h, H)
plt.xlabel("Depth($h$)")
plt.ylabel('The Height of a Wave($H$)')
plt.show()
```

```
plt.plot(X, H)
plt.xlabel("Distance($x$)")
plt.ylabel('The Height of a Wave($H$)')
plt.show()
```

```
N1 = 1/2 * (1 + 2*k0*h0/np.sinh([2*k0*h0]))
```

```
h = []
k = []
C = []
N = []
H = []
Bisection_H = []
```

```
Bisection_H.append(H1)
```

```
H.append(H1)
```

```
h.append(h0)
```

```
bisection_k = []
```

```
bisection_k.append(k0)
```

```
for i in range(0, 99):
    h1 = round(10 - (1/10)*(i+1), 2)
    k_init = sqr_sigma / (g * math.sqrt(np.tanh([sqr_sigma / g * h1])))
    k1 = abs(bisection_method(k_init, sqr_sigma, h1))
    C2 = sigma / k1
    N2 = 1/2 * (1 + 2*k1*h1/np.sinh([2*k1*h1]))
    H2 = H1 * math.sqrt((C1 * N1) / (C2 * N2))
    bisection_k.append(k1)
    h.append(h1)
    k.append(k1)
    C.append(C2)
    N.append(N2)
    H.append(H2)
    Bisection_H.append(H2)
```

```
import matplotlib.pyplot as plt
```

```
X = np.arange(0, 100)
```

```
plt.plot(h, H)
plt.xlabel("Depth($h$)")
plt.ylabel('The Height of a Wave($H$)')
plt.show()
```

```
plt.plot(X, H)
plt.xlabel("Distance($x$)")
plt.ylabel('The Height of a Wave($H$)')
plt.show()
```

```
N1 = 1/2 * (1 + 2*k0*h0/np.sinh([2*k0*h0]))
```

```
h = []
k = []
C = []
N = []
H = []
Eckart_H = []
```

```
Eckart_H.append(H1)
```

```
h.append(h0)
```

```
H.append(H1)
Eckart_H.append(H1)
```

```
eckart_k = []
```

```
eckart_k.append(k0)
```

```
for i in range(0, 99):
    h1 = round(10 - (1/10)*(i+1), 2)
    k1 = sqr_sigma / (g * math.sqrt(np.tanh([sqr_sigma / g * h1])))
    C2 = sigma / k1
    N2 = 1/2 * (1 + 2*k1*h1/np.sinh([2*k1*h1]))
    H2 = H1 * math.sqrt((C1 * N1) / (C2 * N2))
    h.append(h1)
    k.append(k1)
    eckart_k.append(k1)
    C.append(C2)
    N.append(N2)
    H.append(H2)
    Eckart_H.append(H2)
```

```
len(H)
```

```
import matplotlib.pyplot as plt
```

```
X = np.arange(0, 100)
```

```
plt.plot(h, H)
plt.xlabel("Depth($h$)")
plt.ylabel('The Height of a Wave($H$)')
plt.show()
```

```
plt.plot(X, H)
plt.xlabel("Distance($x$)")
plt.ylabel('The Height of a Wave($H$)')
plt.show()
```

```
hunt_k = []
```

```
d = [0.6666666666, 0.3555555555, 0.1608465608, 0.0632098765, 0.0217540484, 0.0076507983]
```

```python
h = []
k = []
C = []
N = []
H = []
Hunt_H = []
```

```python
Hunt_H.append(H1)
```

```python
h.append(h0)
H.append(H1)
```

```python
added_y = []
```

```python
added_y.append(sqr_sigma*h0/g)
```

```python
hunt_k.append(k0)
```

```python
for i in range(0, 99):
    h1 = round(10 - (1/10)*(i+1), 2)

    y = sqr_sigma*h1/g
    added_y.append(y)
    dny = 0

    for j in range(0, len(d)):
        dny += d[j] * (y ** (j+1))

    k1 = math.sqrt(((y) ** 2 + ((y) / (1 + dny))) / (h1 ** 2))
    hunt_k.append(k1)

    C2 = sigma / k1
    N2 = 1/2 * (1 + 2*k1*h1/np.sinh([2*k1*h1]))
    H2 = H1 * math.sqrt((C1 * N1) / (C2 * N2))
    h.append(h1)
    k.append(k1)
    C.append(C2)
    N.append(N2)
    H.append(H2)
    Hunt_H.append(H2)

```

```python
abs_Eckart_H = 0
abs_Hunt_H = 0
```

```python
for i in range(0, len(Bisection_H)):
    abs_Eckart_H += abs(Bisection_H[i] - Eckart_H[i])
    abs_Hunt_H += abs(Bisection_H[i] - Hunt_H[i])
```

```python
mean_abs_Eckart_H = abs_Eckart_H / len(Bisection_H)
mean_abs_Hunt_H = abs_Hunt_H / len(Bisection_H)
```

```python
mean_abs_Eckart_H * 100
```

```python
mean_abs_Hunt_H * 100
```

```python
len(hunt_k)
```

```python
len(k)
```

```python
import matplotlib.pyplot as plt
import numpy as np
import math

# Plotting both the curves simultaneously
plt.plot(h, bisection_k, color='r', label='Bisection Method')
plt.plot(h, eckart_k, color='b', label='Eckart')
plt.plot(h, hunt_k, color='g', label='Hunt')

# Naming the x-axis, y-axis and the whole graph
plt.xlabel("Depth($h$)")
plt.ylabel("Wave Number($k$)")
plt.title("Wave Number($k$) vs. Depth($h$)")

# Adding legend, which helps us recognize the curve according to it's color
plt.legend()
plt.xlim(0, 1)

# To load the display window
plt.show()
```

```python
abs_Hunt_k = 0
abs_Eckart_k = 0
```

```python
for i in range(0, len(hunt_k)):
    abs_Hunt_k += abs(hunt_k[i] - bisection_k[i])
    abs_Eckart_k += abs(eckart_k[i] - bisection_k[i])
```

```
mean_abs_Hunt_k = abs_Hunt_k / len(bisection_k)
mean_abs_Eckart_k = abs_Eckart_k / len(bisection_k)
```

```
mean_abs_Hunt_k * 100
```

```
mean_abs_Eckart_k * 100
```

```
import matplotlib.pyplot as plt
```

```
X = np.arange(0, 100)
```

```
plt.plot(h, H)
plt.xlabel("Depth($h$)")
plt.ylabel('The Height of a Wave($H$)')
plt.show()
```

```
plt.plot(X, H)
plt.xlabel("Distance($x$)")
plt.ylabel('The Height of a Wave($H$)')
plt.show()
```

```
Bisection_C = []
Eckart_C = []
Hunt_C = []
```

```
for i in range(0, len(bisection_k)):
    Bisection_C.append((2 * math.pi / bisection_k[i])/T0)
```

```
for i in range(0, len(eckart_k)):
    Eckart_C.append((2 * math.pi / eckart_k[i])/T0)
```

```
for i in range(0, len(hunt_k)):
    Hunt_C.append((2 * math.pi / hunt_k[i])/T0)
```

```
import matplotlib.pyplot as plt
import numpy as np
import math

# Plotting both the curves simultaneously
plt.plot(h, Bisection_C, color='r', label='Bisection Method')
plt.plot(h, Eckart_C, color='g', label='Eckart')
plt.plot(h, Hunt_C, color='b', label='Hunt')

# Naming the x-axis, y-axis and the whole graph
```

```
plt.xlabel("Depth($h$)")
plt.ylabel("$C$")
plt.title("Wave Velocities($C$) vs. Depth($h$)")

# Adding legend, which helps us recognize the curve according to it's color
plt.legend()

# To load the display window
plt.show()
```

```
abs_Eckart_C = 0
abs_Hunt_C = 0
```

```
for i in range(0, len(Bisection_C)):
    abs_Eckart_C += abs(Bisection_C[i] - Eckart_C[i])
    abs_Hunt_C += abs(Bisection_C[i] - Hunt_C[i])
```

```
mean_abs_Eckart_C = abs_Eckart_C / len(Eckart_C)
mean_abs_Hunt_C = abs_Hunt_C / len(Hunt_C)
```

```
mean_abs_Eckart_C * 100
```

```
mean_abs_Hunt_C * 100
```

```
import matplotlib.pyplot as plt
import numpy as np
import math

# Plotting both the curves simultaneously
plt.plot(h, added_y, color='r', label='The y value of the Hunt method.')

# Naming the x-axis, y-axis and the whole graph
plt.xlabel("Index")
plt.ylabel("$y$")
plt.title("y value")

ymax = max(added_y)

s = 'The maximum value of y= ' + str(ymax) + ' '

plt.annotate(s, (3, 0))

# Adding legend, which helps us recognize the curve according to it's color
```

```python
plt.legend()

# To load the display window
plt.show()
```

```python
ymax = max(added_y)
```

```python
ymax
```