ISSN 2321 3361 © 2019 IJESC

Research Article Volume 9 Issue No.11

Computational Tool for the Characteristics of Linear Water Waves using C# Language

Mohamed Abouelnasr[1], Mohamed Tarek[2], ZiadSliman[3]
Port Said University, Port Said, Egypt

Abstract:
During the past centuries, there were many attempts to find the water wave characteristics. Knowledge of the water wave characteristics is essential for the design of coastal structures and coastal utilities in the civil and coastal engineering. In the past studies on the formation and characteristics of the regular water waves, the objective was to provide a detailed understanding of the mechanics of a wave field through examination of waves of constant height and period. The simplest wave theory for regular water waves is the first-order, small-amplitude, Airy wave theory, or linear regular wave theory. In this work, we have described the linear water waves and their parameters. We have developed a computational tool for calculating and plotting the linear water wave characteristics such as water wavelength and water wave celerity using C# language.

Keywords: Airy waves, Computational tool, linear water waves, small-amplitude, water wave theory.

I. INTRODUCTION

Linear wave theory or Airy Waves deals with the simplest mathematical representation assuming ocean waves are two-dimensional (2-D), small in amplitude, and progressively definable by their wave height and period in a given water depth[1]-[2].We have developed a computational tool for calculating the linear water wave characteristics as per the linear wave theory or Airy theory using C# language.

II. DEFINITION OF LINEAR WAVE PARAMETERS

A progressive wave may be represented by the variables x (spatial) and t (temporal) or by their combination (phase), defined as θ = kx - ωt, where k and ω are described below and the values of θ vary between 0 and 2π[2]. A simple, periodic wave of permanent form propagating over a horizontal bottom may be completely characterized by the wave height "H", wavelength "L", and water depth "d" as per below figure 1.

- H is the wave height
- Tis the wave period
- ω = 2π/T is the angular or radian frequency,
- k = 2π/L is the wave number,
- C = L/T = ω/k is The phase velocity or wave celerity,

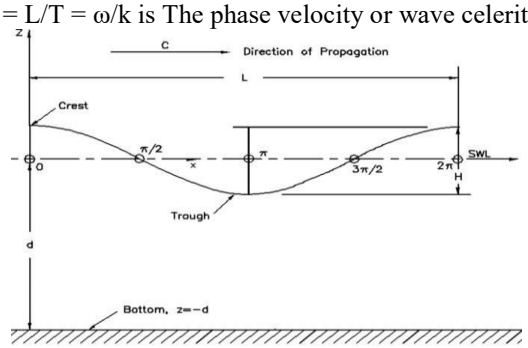

Figure.1.Progressive Linear Water Wave [2]

The expression relating wave celerity to wavelength and water depth and wavelength as a function of depth and wave is obtainedas per the below equation, refer to (1),

$$C = (g \times T) / (2\pi) \times \tanh((2\pi d) / L) \quad (1)\ [1]$$

The expression relating the wavelength as a function of depth and wave period is obtainedas per the below equation, refer to (2).

$$L = (g \times T^2) / (2\pi) \times \tanh((2\pi d) / L) \quad (2)\ [1]$$

The method of searching for a solution of an unknown value follows a trial and error approach. Trial and error is a basic methodology in problem solving and knowledge acquisition [3]. The expression relating the expression for approximate value of the wavelength as a function of depth and wave period[4] is obtainedas per the below equation, refer to (3).

$$Lo = ((g \times T^2)/(2\pi)) \times \tanh((2\pi d) / L) \quad (3)\ [4]$$

III. COMPUTATIONAL TOOL CODE USING C# LANGUAGE

The computational tool's form was been designed and written by C# language. C# is a simple, modern, object-oriented, and type-safe programming language.C# is an object-oriented language, but C# further includes support for component-oriented programming.C# has its roots in the C family of languages [5].The computational tool's form is named "LWAVE" and it is shown below as per figure 2.

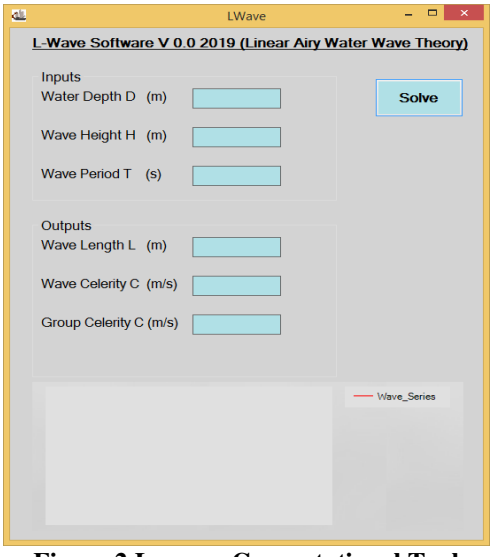

Figure.2.Lwave - Computational Tool

The computational tool's C# code is shown below as per figure 3,figure 4 and figure 5.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace LWave
{
public partial class LWave : Form
{
public LWave()
{InitializeComponent();}
private void Solve_Click(object sender,
EventArgs e)
{// list of Abbreviations:
// SWL = Sea Water Level
// D  = water depth measured from SWL
// H  = Wave Height
// L  = Wavelength
// T  = Wave Period
// C  = Wave Celerity
// Cg = Group Wave Celerity
// RD = Relative Depth
// g  = Global Gravitational Acceleration
Double g;
g = 9.81;
Double pi;
pi = Math.PI;
Double D;
D = double.Parse(D_textBox.Text);
Double H;
H = double.Parse(H_textBox.Text);
Double T;
T = double.Parse(T_textBox.Text);
Double L;
```

Figure.3.Lwave - Computational Tool's C# Code 1/3

```
Double C;
Double Cg;
Double L_Eckhart;
L_Eckhart = Math.Round((g * T * T / (2 * pi)) *
Math.Sqrt(Math.Tanh((4 * pi * pi * D) / (T * T *
g))),5);
Double Lc;
Lc = Math.Round(((g * T * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L_Eckhart)), 5);
Double M_error;
M_error = L_Eckhart - Lc;
Double L0_0, Lc_0, Error_0 ;
// L0_0 = L_Eckhart , Lc_0 = Lc , Error_0 =
M_error
L0_0 = L_Eckhart;
Lc_0 = Lc;
Error_0 = M_error;
double L0_1, Lc_1, Error_1;
double L0_2, Lc_2, Error_2;
double L0_3, Lc_3, Error_3;
double L0_4, Lc_4, Error_4;
double L0_5, Lc_5, Error_5;
double L0_6, Lc_6, Error_6;
double L0_7, Lc_7, Error_7;
double L0_8, Lc_8, Error_8;
double L0_9, Lc_9, Error_9;
// First Try & Error step for solving the
equations
if (Math.Abs(Error_0) <= 0.010)
{L_textBox.Text = Lc_0.ToString();
L = Lc_0;}
else
{// Second Try & Error step for solving the
equations
L0_1 = L0_0 - Error_0 / 2;
Lc_1 = Math.Round(((g * T * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L0_1)), 5);
Error_1 = L0_1 - Lc_1;
if (Math.Abs(Error_1) <= 0.010)
{L_textBox.Text = Lc_1.ToString();
L = Lc_1;}
else
{// Third try & Error step for solving the
equations
L0_2 = L0_1 - Error_1 / 2;
Lc_2 = Math.Round(((g * T * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L0_2)), 5);
Error_2 = L0_2 - Lc_2;
if (Math.Abs(Error_2) <= 0.010)
{L_textBox.Text = Lc_2.ToString();
L = Lc_2;}
else
{// Fourth try & Error step for solving the
equations
L0_3 = L0_2 - Error_2 / 2;
Lc_3 = Math.Round(((g * T * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L0_3)), 5);
Error_3 = L0_3 - Lc_3;
if (Math.Abs(Error_3) <= 0.010)
{L_textBox.Text = Lc_3.ToString();
L = Lc_3;}
else
{// Fifth try & Error step for solving the
equations
L0_4 = L0_3 - Error_3 / 2;
Lc_4 = Math.Round(((g * T * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L0_4)), 5);
Error_4 = L0_4 - Lc_4;
if (Math.Abs(Error_4) <= 0.010)
{L_textBox.Text = Lc_4.ToString();
L = Lc_4;}
else
{// Sixth try & Error step for solving the
equations
```

Figure.4.Lwave - Computational Tool's C# Code 2/3

```
L0_5 = L0_4 + Error_4 / 2;
Lc_5 = Math.Round(((g * T * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L0_5)), 5);
Error_5 = L0_5 - Lc_5;
if (Math.Abs(Error_5) <= 0.010)
{L_textBox.Text = Lc_5.ToString();
L = Lc_5;}
else
{// Seventh try & Error step for solving equations
L0_6 = L0_5 - Error_5 / 2;
Lc_6 = Math.Round(((g * T * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L0_6)), 5);
Error_6 = L0_6 - Lc_6;
if (Math.Abs(Error_6) <= 0.010)
{L_textBox.Text = Lc_6.ToString();
L = Lc_6;}
else
{// Eighth try & Error step for solving equations
L0_7 = L0_6 - Error_6 / 2;
Lc_7 = Math.Round(((g * T * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L0_7)), 5);
Error_7 = L0_7 - Lc_7;
if (Math.Abs(Error_7) <= 0.010)
{L_textBox.Text = Lc_7.ToString();
L = Lc_7;}
else
{// Ninth try & Error step for solving equations
L0_8 = L0_7 - Error_7 / 2;
Lc_8 = Math.Round(((g * T * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L0_8)), 5);
Error_8 = L0_8 - Lc_8;
if (Math.Abs(Error_8) <= 0.010)
{L_textBox.Text = Lc_8.ToString();
L = Lc_8;}
else
{// tenth try & Error step for solving equations
L0_9 = L0_8 - Error_8 / 2;
Lc_9 = Math.Round(((g * T * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L0_9)), 5);
Error_9 = L0_9 - Lc_9;
L_textBox.Text = Lc_9.ToString();
L = Lc_9;}}}}}}}}}
// Water Wave Celerity C & Group Water Wave
Celerity Cg
Double RD;
RD = D / L;
C = Math.Round((g * T / (2 * pi)) * Math.Tanh(2 *
pi * D / L), 5);
C_textBox.Text = C.ToString();
if (Math.Abs(RD) >= 0.5)
{Cg = Math.Round(((g * T) / (4 * pi)), 5);
Cg_textBox.Text = Cg.ToString();
Wave_Area_label.Text = "Type of Water Is -Deep
Water-";}
if (Math.Abs(RD) <= 0.04)
{Cg = Math.Round((g * T / (2 * pi)) * Math.Tanh(2
* pi * D / L), 5);
Cg_textBox.Text = Cg.ToString();
Wave_Area_label.Text = "Type of Water Is -Shallow
Water-";}
if (Math.Abs(RD) > 0.04 && Math.Abs(RD) < 0.5)
{double V = 4 * pi * D / L;
Cg = Math.Round(((Math.Sinh(V) + V) / (2 *
Math.Sinh(V))) * Math.Round((g * T / (2 * pi)) *
Math.Tanh(2 * pi * D / L), 5), 5);
Cg_textBox.Text = Cg.ToString();
Wave_Area_label.Text = "Type of Water Is
-Transitional Water-";}
// Plotting wave characteristics
double X,double Y;
for (X = 0; X <= L; X++)
{Y = (H / 2) * Math.Cos(2 * pi * X / L);
Wave.Series["Wave_Series"].Points.AddXY(X,
Y);}}}}
```

Figure.5.Lwave - Computational Tool's C# Code 3/3

The result of the executing C# code of the "LWAVE" computational tool for the inputs of water depth equals to 150 m/sec, wave period equals to 10 sec and wave height equals to 3 m is shown as per figure 6.

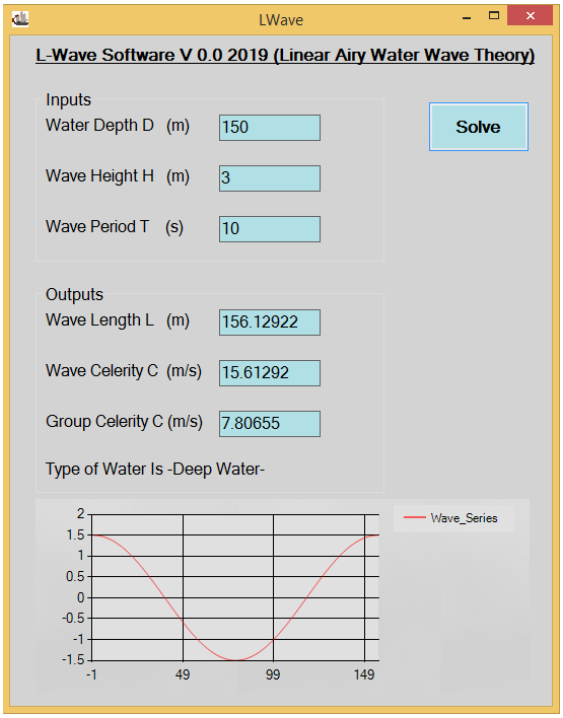

Figure.6.Lwave –Sample Result Of Computational Tool

IV. CONCLUSION

The computational tool "LWAVE" for the linear wave theory is able to calculate the wave characteristics such as water wavelength, wave celerity and waves' group celerity with percentage of error less than 1% of the water wavelength. The tool can be used for deep water waves, transitional water waves and shallow water waves.

V. ACKNOWLEDGEMENT

We like to show our gratitude to Mr.Juma Obaid Mubarak, managing director of Mubarak Marine L.L.C for sharing his pearls of wisdom with us during the preparation of this researchand his continues support.

V. REFERENCES

[1].G. Airy. (1845). "Tides and Waves, Encyc. Metrop". Article 192.

[2]. Z. Demirbilek and C. Linwood. (2002, April). "Coastal Engineering Manual 1110-2-1100 (Part II)". [Online]. Available: https://publications.usace.army.mil/USACE-Publications/Engineer-Manuals/u43544q/636F617374616C20656 E67 696E656572696E67206D616E75616C/

[3].X. Bei, N. Chen and S.Zhang (2015). "Trial and Error Algorithms, Encyclopedia of Algorithms". Springer Article.

[4]. C. Eckart. (1952). "The Propagation of Gravity Waves from Deep to Shallow Water". Circular 521, pp 165-173.

[5]. Microsoft. (2012). "C# Language Specification Version 5 - 2-1100 (Part II)". [Online].Available: https: //docs. Micro soft. com/ en- us/ dotnet/ csharp/language-reference/language-specifi ca tion/