https://github.com/ARPA-GO/shipcurve

| Python based ship calculation open source library

ARPA-GO

Always Ready for Program Assistance - GO

COPYRIGHT

LICENSE NOTICE

PYTHON

Version : Python 3.7.3

Download : https://docs.python.org/3/download.html

DOCUMENTS

INTRODUCTION

This document explains how to install and use ARPA-GO.

READERS

The readers of this document will be users of platform "github" who are interested in ship calculation especially hydrostatic curves.

CONTACT INFORMATION

If there is an error or have any questions regarding this content, please contact us.
E-mail: ARPAGO.kmou@gmail.com

DOCUMENTS VERSION

VERSION	DATE	HISTORY
1.0	2019.05.20	1.0 Distribution

INDEX

1. ARPA-GO Introduction

This chapter is all about ARPA-GO.

1.1 What is ARPA-GO?

Always Ready for Program Assistance

It's the full form of our project 'ARPA-GO'. There are already many commercial shipping programs such as Foran, Aveva Marine. But students and Start-up Company find it difficult to use these programs because of their high prices. And also we cannot easily use the program before it is educated.

To solve this problem, we developed a library. ARPA-GO is python based ship calculation open source library. Ship calculations are divided into various fields. Among them, hydrostatic curve which is based on all ship calculations was prioritized and also provided ship calculation sheet for additional calculation.

1.2 Functions and Characteristics

The functions and features of ARPA-GO are as follows:

Open source

As mentioned before, ARPA-GO is python based ship calculation open source. It provides free access to anyone who wants to use. You can find this project in the platform 'github'. Users can find the information they want without any burden and also share opinions with other users.

GitHub is a web-based hosting service for version control using Git. It provides access control and several collaboration features such as feature requests, task management for every project. Open source development of activities in various fields is underway on this platform. On the other hand, the development in shipping sector was difficult to find. The big objective of the project is to promote open source development in the shipping sector through ARPA-GO.

One click touch

Project ARPA-GO progresses ship calculation by using offset table of ship. The results of this calculation are hydrostatic curves and calculation table of ship which is about coefficient, KB, moment of ship. To calculate ship, there are many variables required such as breadth, distance between stations, length between perpendiculars and the number of waterline, etc. We import all these variables from offset directly. As a result, we don't have any variables to be entered for ship calculation. With no variables, our project performs complex calculations. This means only offset table is required to calculate ship in our project ARPA-GO.

Two forms

ARPA-GO is composed of two types. One is the list of codes that are convenient to modify as needed, and the other is the GUI format which is easy to execute. Both types are accessible. Users can choose their own preference. Chapter 3 will cover two forms in detail.

CAD: Coordinate extraction

For ship calculation, offset which is divided by each station and waterline of ship is necessary. This offset is usually extracted manually from the body plan of ship. On the other hand, ARPA-GO automatically accepts offset and principle dimensions through lisp. It is saved as Excel file (*.xlsx). From Python, import the completed offset through the command and start calculation. We will cover this function in more detail in the appendix.

Comparison value

In the process of ship design, first we conduct ship calculation of mothership. And then changes the values to compare them with the mothership's value to find the optimal model. ARPA-GO has added a function to compare the values of the existing and revised ship. This function enables efficient comparison of results. Let us discuss about it later.

2. Installation

This chapter describes the two ways to use ARPA-GO process.

2.1 Pycharm

Since ARPA-GO is python based, users should install Python except for case of using exe file of ARPA-GO. Python is an interpreted, high-level, general-purpose programming language. Its language constructs and object-oriented approach aims to help programmers write clear, logical code for small and large-scale projects. This is why Python is used as our library development tool.

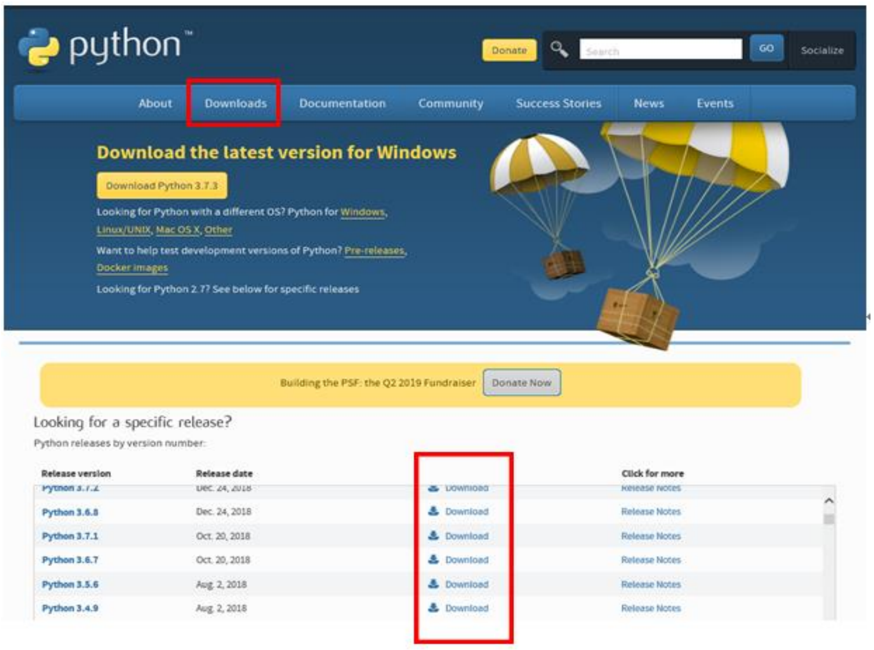
The Python installation process and process verification are included below, along with detailed figures.

2.1.1 Python Installation

1. Access the official website of Python. The official website URL is https://www.python.org/ .

   You can read more information about this computer language here.

2. Click Download from the menu at the top to display various versions of Python files. Click the desired version of Download to save the file. ARPA-GO is using Python version of 3.7.3.



Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Pre-releases, Docker images

Looking for Python 2.7? See below for specific releases

Building the PSF: the Q2 2019 Fundraiser

Looking for a specific release?

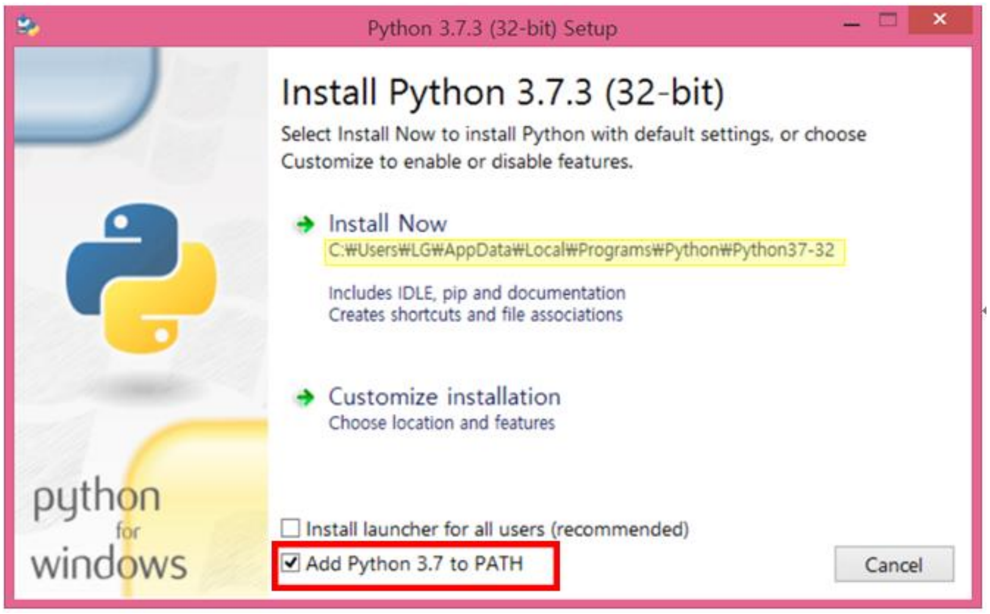
Python releases by version number:

Release version	Release date		Click for more
Python 3.7.2	Dec. 24, 2018	Download	Release Notes
Python 3.6.8	Dec. 24, 2018	Download	Release Notes
Python 3.7.1	Oct. 20, 2018	Download	Release Notes
Python 3.6.7	Oct. 20, 2018	Download	Release Notes
Python 3.5.6	Aug. 2, 2018	Download	Release Notes
Python 3.4.9	Aug. 2, 2018	Download	Release Notes

Notice

Since Python 2.x will be supported by 2020, we recommend version python 3.x

3. Install the downloaded file and you will see the following screen. Check **'Add Python 3.7 to PATH'** and click **'Install Now'** to proceed. If you want to change the location of downloaded file, select **'Customize Installation'**.



Python 3.7.3 (32-bit) Setup

Install Python 3.7.3 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

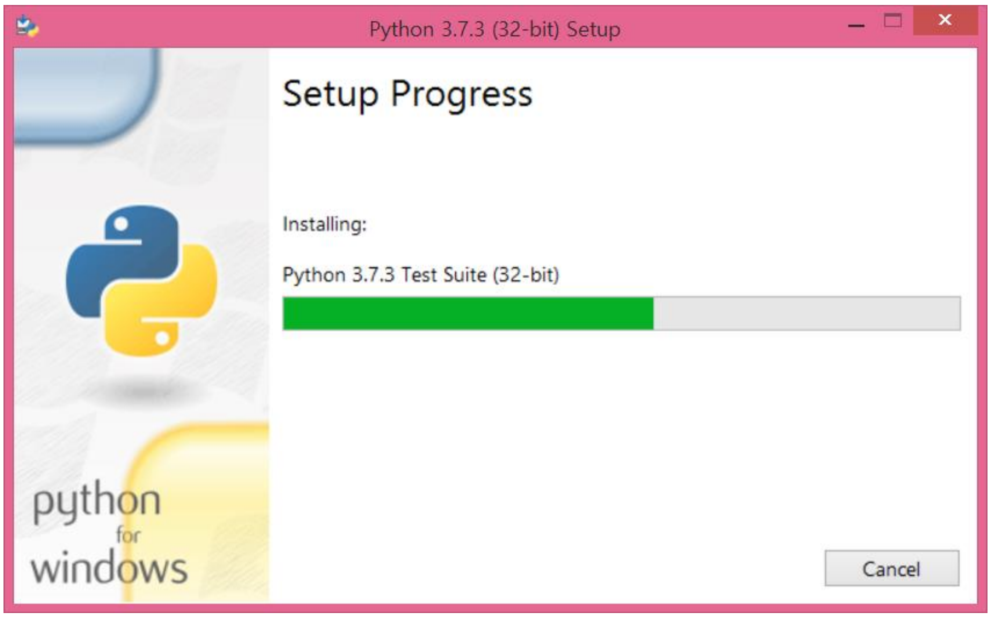
Install Now
C:₩Users₩LG₩AppData₩Local₩Programs₩Python₩Python37-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☐ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

4. Installation is in progress.



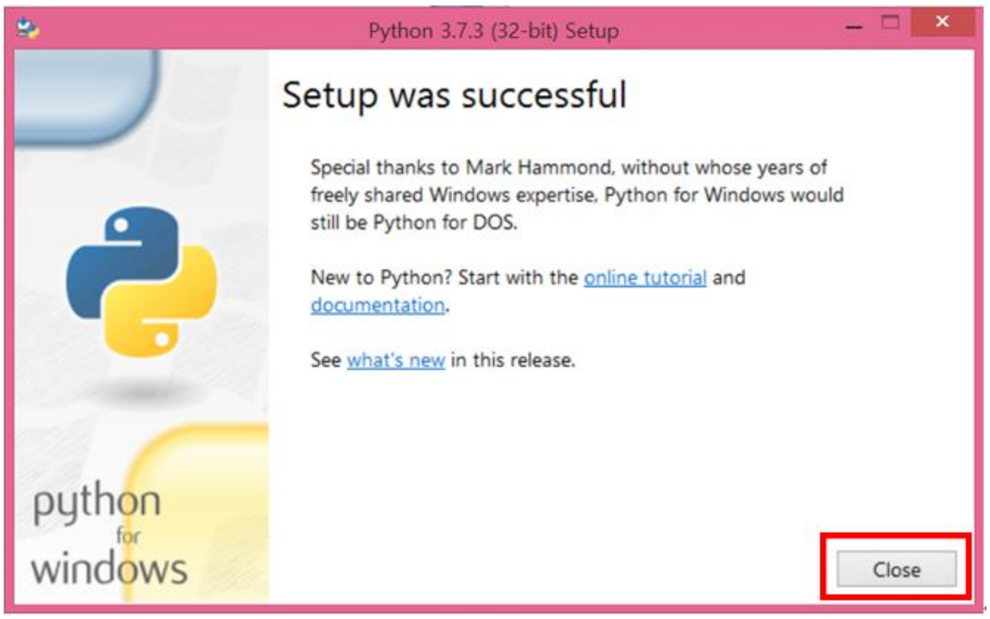
Python 3.7.3 (32-bit) Setup

Setup Progress

Installing:

Python 3.7.3 Test Suite (32-bit)

5. Click **'Close'** to complete the installation.



Setup was successful

Special thanks to Mark Hammond, without whose years of freely shared Windows expertise, Python for Windows would still be Python for DOS.

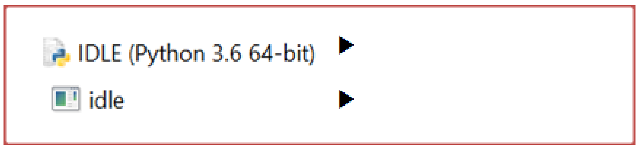
New to Python? Start with the online tutorial and documentation.

See what's new in this release.

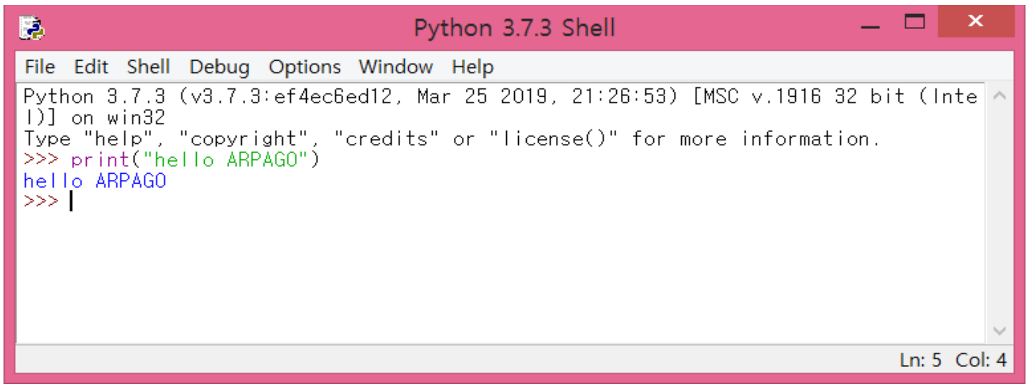
2.1.2 Python Execution

To execute Python: Start menu → recently added application → IDLE (Python 3.7)

If there are no recently added app items, using 'Search for Programs and Files' execute IDLE.

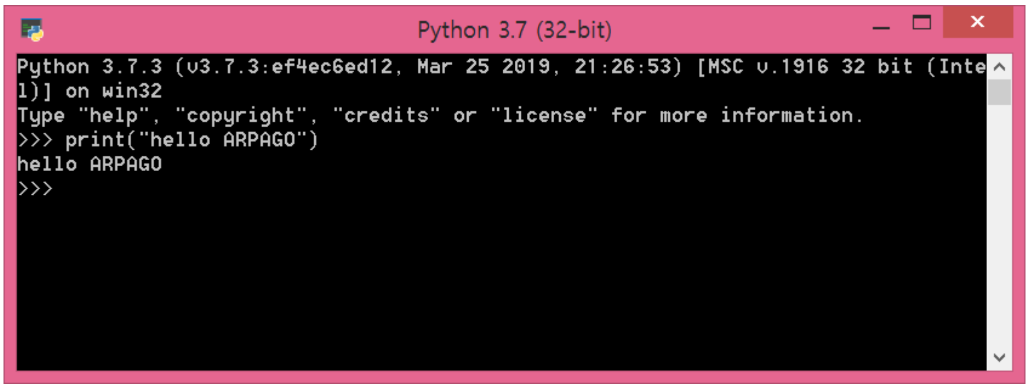


'hello ARPAGO' has been printed on Shell to confirm execution.



```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("hello ARPAGO")
hello ARPAGO
>>> |
```

'hello ARPAGO' has been printed on Console to confirm execution.



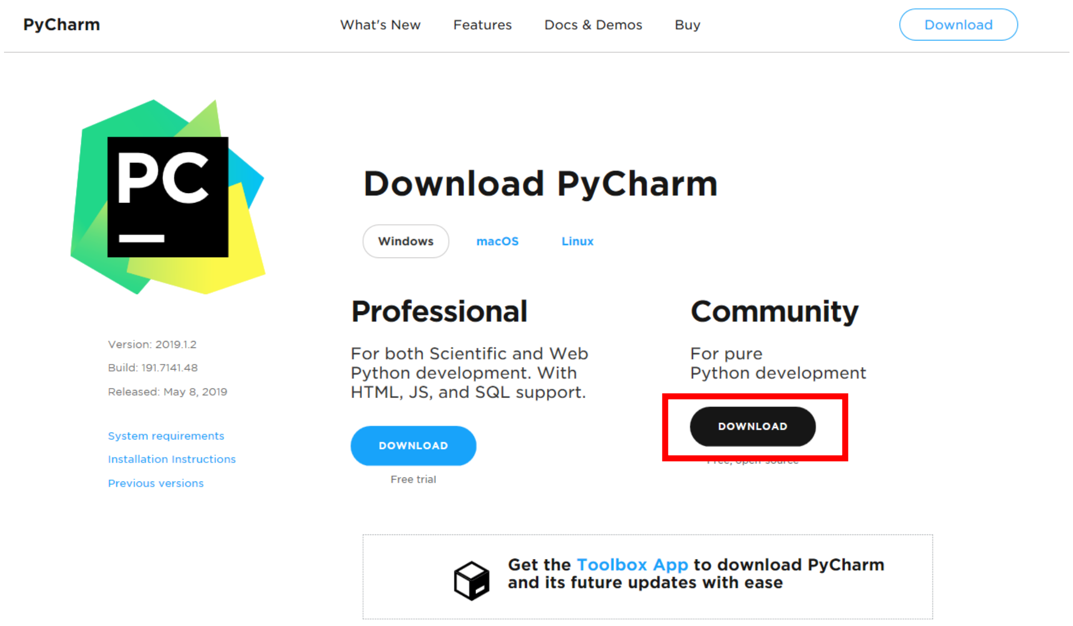
```
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53) [MSC v.1916 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello ARPAGO")
hello ARPAGO
>>>
```

2.1.3 pycharm Installation

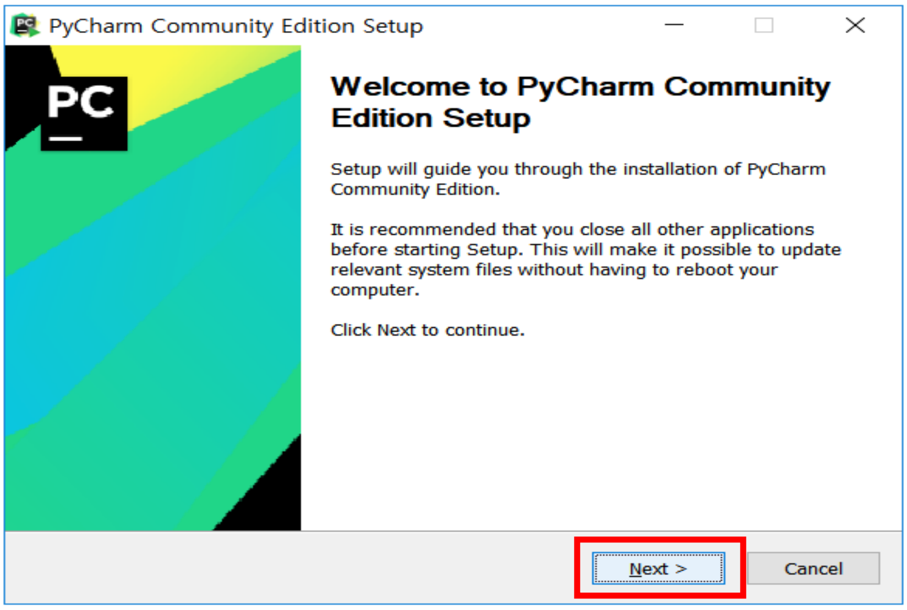
There are various integration environments in which you can develop python. Among them, We describe about the installation and execution method of pycharm. Unlike IDLE window of Python, Pycharm provides version selection and the execution of packages is easy. Also, it is easy to identify the file and function of python codes. Not only that it is very convenient to manage.

1. Access the official website of Pycharm. The official website URL is https://www.jetbrains.com/pycharm/download/#section=windows.

   Pycharm has a version of professional and community. Since the community version is free, users can download the community version.



2. Run Pycharm installation file and click [Next] button.


PyCharm Community Edition Setup

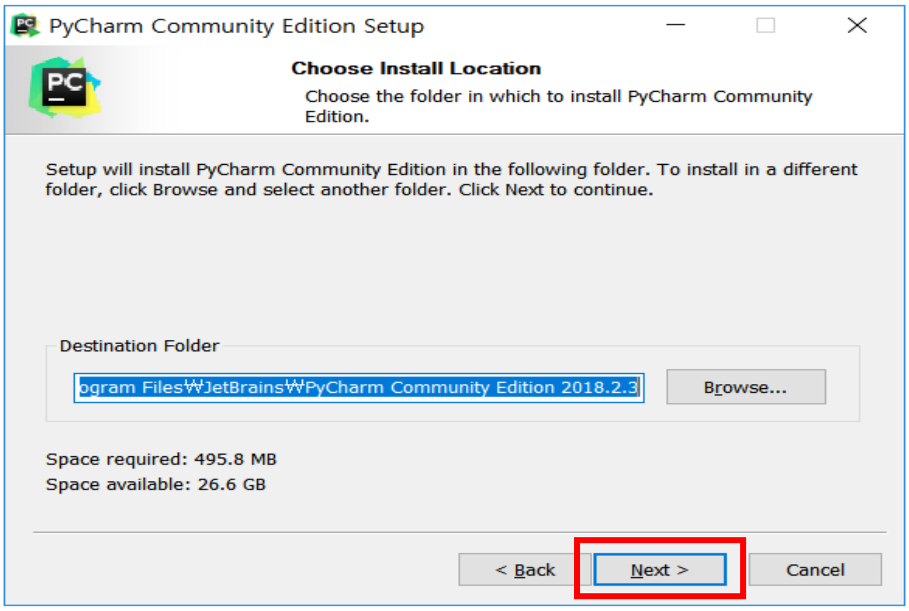
Welcome to PyCharm Community Edition Setup

Setup will guide you through the installation of PyCharm Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

3. Set the Pycharm installation path. If you want to change the installation path, you can change it. If you don't want to change, press the [Next] button directly.


PyCharm Community Edition Setup

Choose Install Location

Choose the folder in which to install PyCharm Community Edition.

Setup will install PyCharm Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

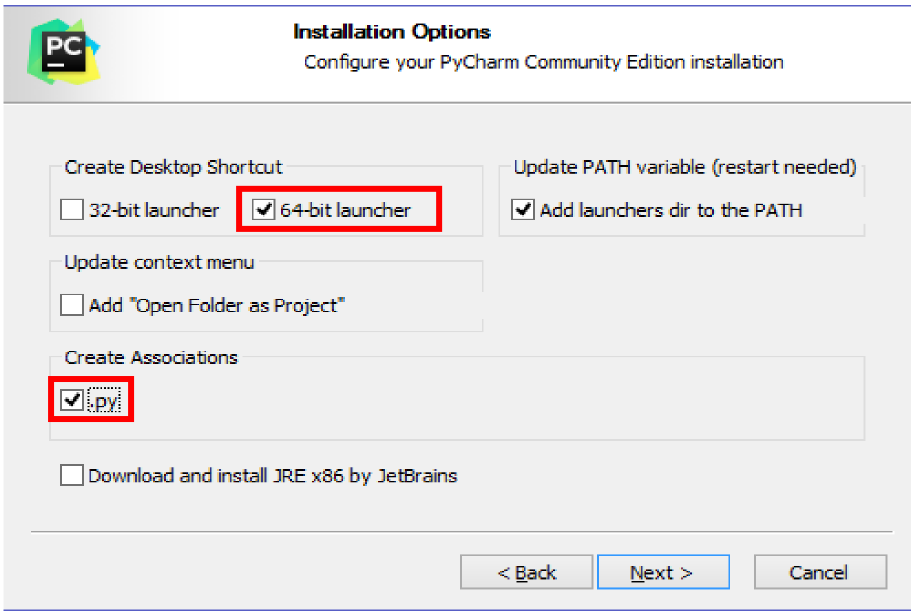
Destination Folder

ogram Files₩JetBrains₩PyCharm Community Edition 2018.2.3

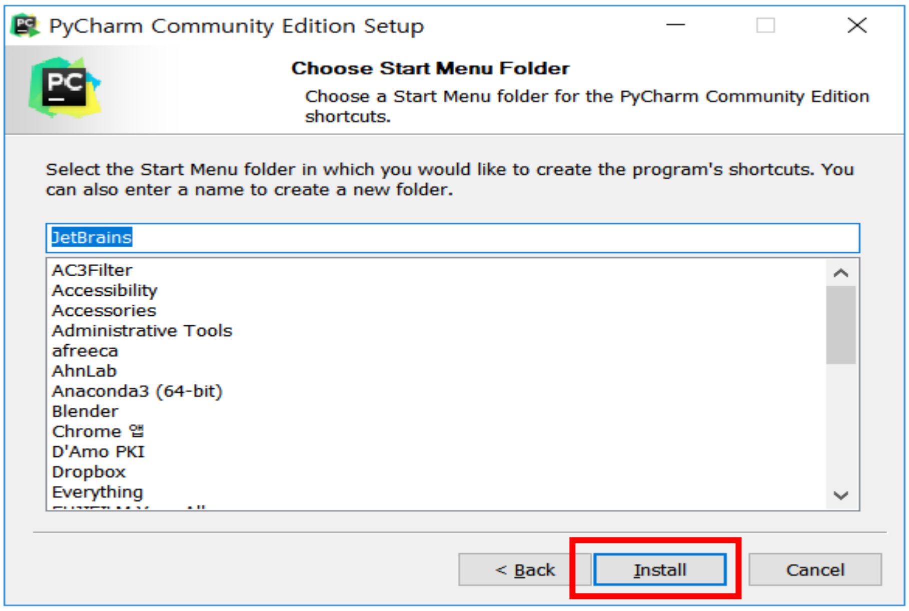
Space required: 495.8 MB
Space available: 26.6 GB

4. Select the installation options. You have to select whether you will create desktop shortcut, update path variable, context menu, and associations.

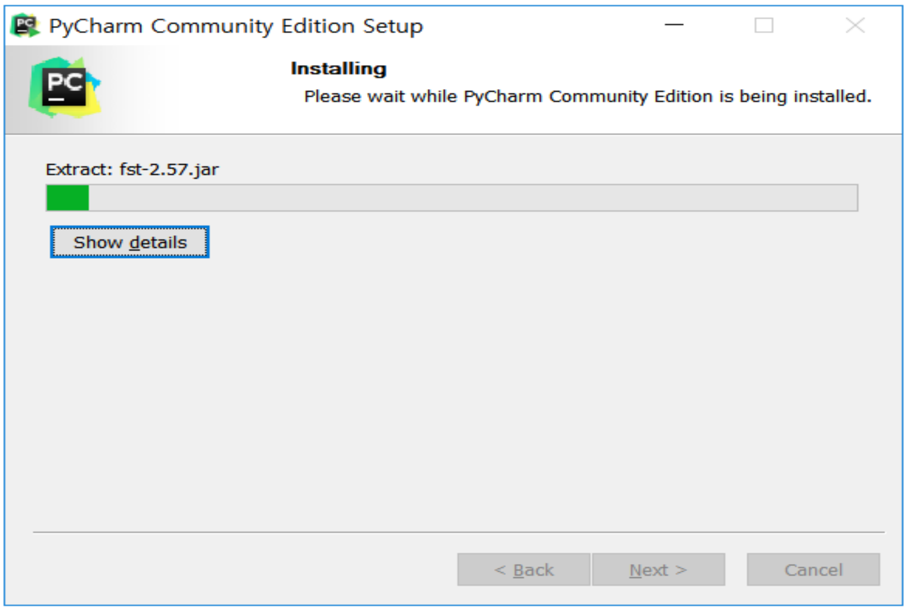
In the section of 'Create Desktop Shortcut' is recommended to check after confirmingr the current window's bit.


Installation Options
Configure your PyCharm Community Edition installation
Create Desktop Shortcut
☐ 32-bit launcher
☑ 64-bit launcher
Update PATH variable (restart needed)
☑ Add launchers dir to the PATH
Update context menu
☐ Add "Open Folder as Project"
Create Associations
☑ .py
☐ Download and install JRE x86 by JetBrains
< Back
Next >
Cancel

5. This screen is to select the Start menu folder. Press the [Install] button without any setting.


PyCharm Community Edition Setup
Choose Start Menu Folder
Choose a Start Menu folder for the PyCharm Community Edition shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
JetBrains
AC3Filter
Accessibility
Accessories
Administrative Tools
afreeca
AhnLab
Anaconda3 (64-bit)
Blender
Chrome 앱
D'Amo PKI
Dropbox
Everything
< Back
Install
Cancel

6. Installation is in progress. It doesn't take much time.

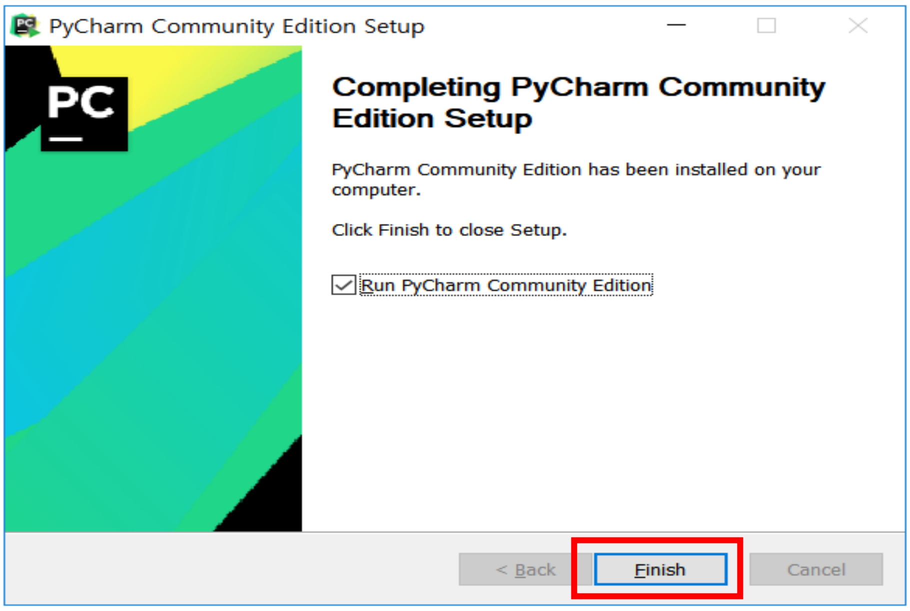

PyCharm Community Edition Setup

Installing

Please wait while PyCharm Community Edition is being installed.

Extract: fst-2.57.jar

7. Check **Run Pycharm Community Edition** and complete the installation..


PyCharm Community Edition Setup

Completing PyCharm Community Edition Setup

PyCharm Community Edition has been installed on your computer.

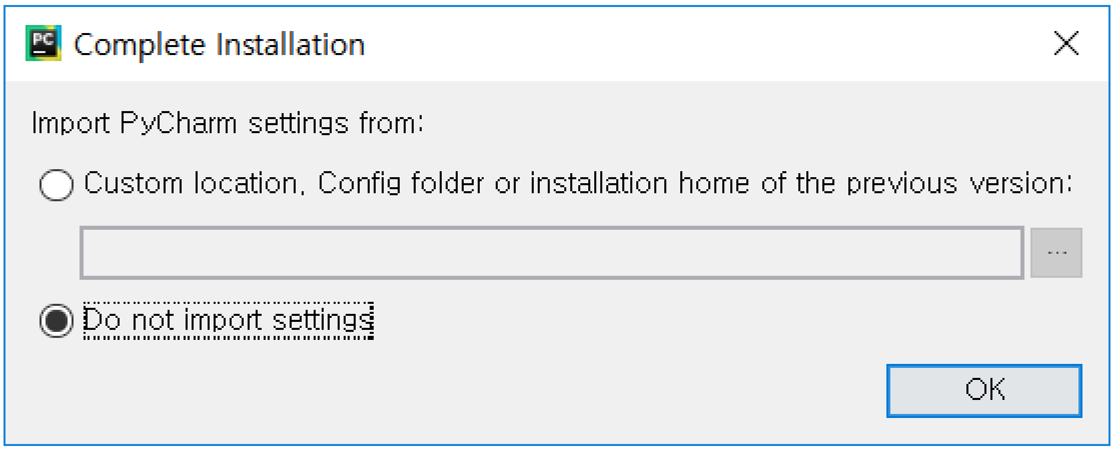
Click Finish to close Setup.

☑ Run PyCharm Community Edition

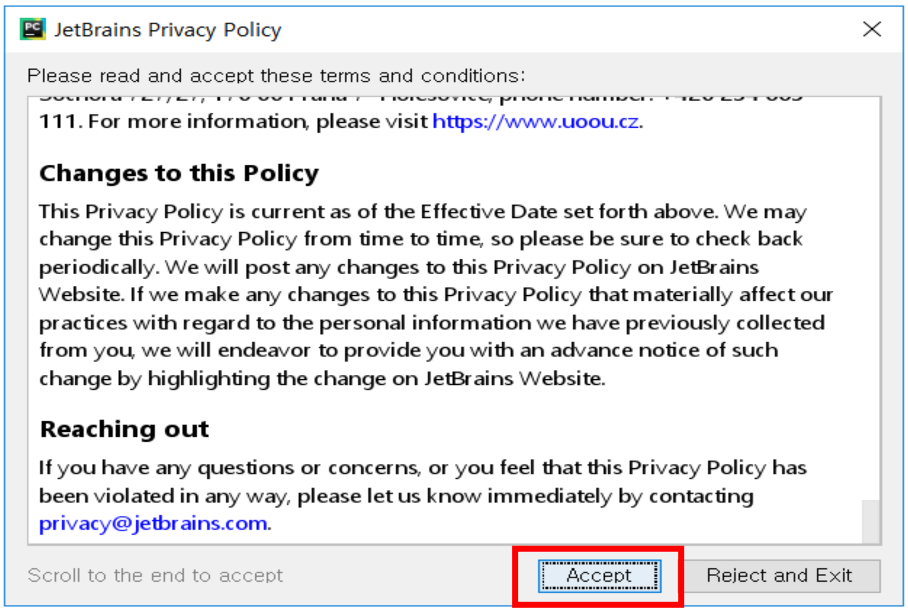
Finally, Pycharm installation is done. Run Pycharm to set up your environment.

8. If you run Pycharm successfully, the following screen will appear.

Select **Do not import settings**, and then click [OK] button.


Complete Installation
Import PyCharm settings from:
Custom location. Config folder or installation home of the previous version:
[✓ Do not import settings]
OK

9. Then click **continue** button. The screen below is a window to accept the JetBrains Privacy Policy. Check **Accept** and continue.



JetBrains Privacy Policy

Please read and accept these terms and conditions:

111. For more information, please visit https://www.uoou.cz.

Changes to this Policy

This Privacy Policy is current as of the Effective Date set forth above. We may change this Privacy Policy from time to time, so please be sure to check back periodically. We will post any changes to this Privacy Policy on JetBrains Website. If we make any changes to this Privacy Policy that materially affect our practices with regard to the personal information we have previously collected from you, we will endeavor to provide you with an advance notice of such change by highlighting the change on JetBrains Website.

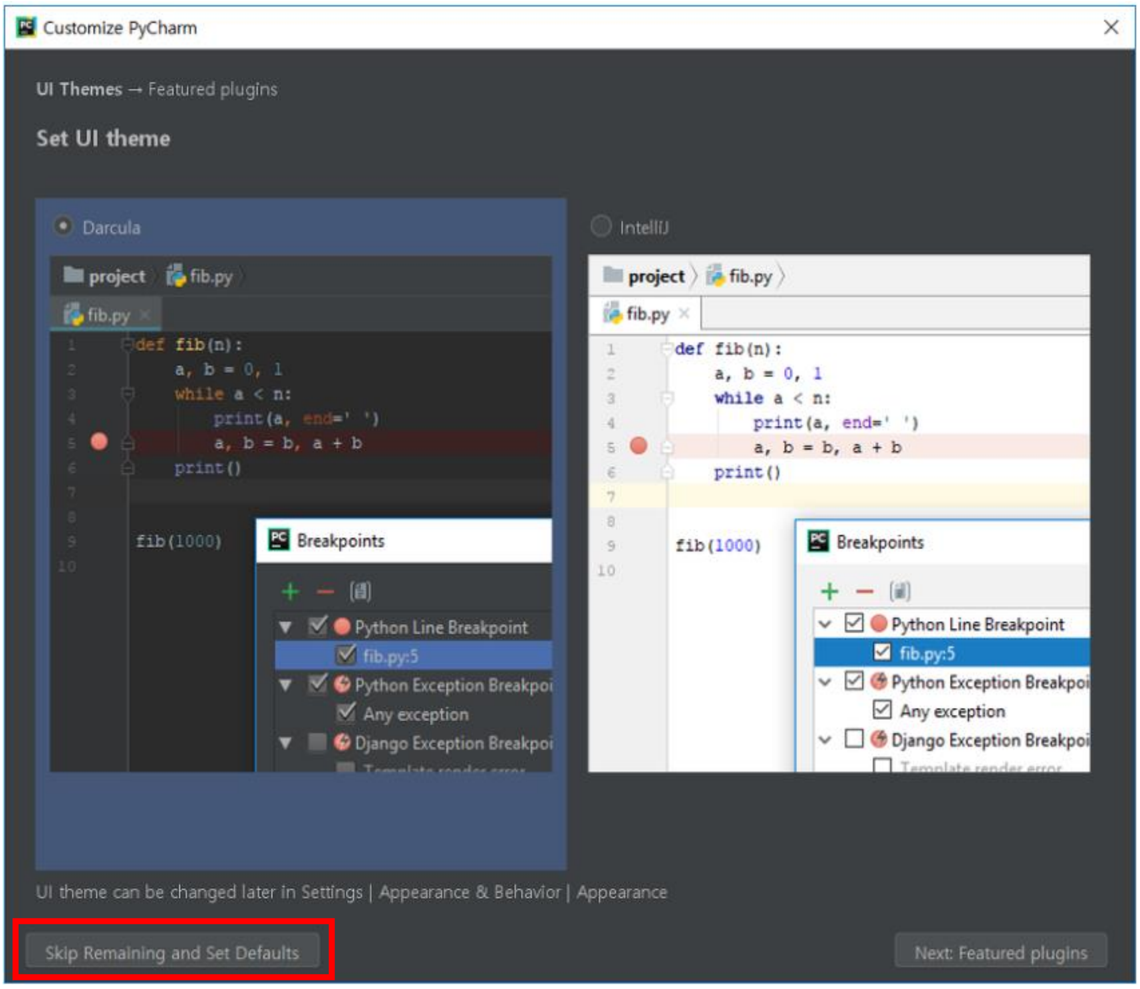
Reaching out

If you have any questions or concerns, or you feel that this Privacy Policy has been violated in any way, please let us know immediately by contacting privacy@jetbrains.com.

10. Select the Pycharm UI theme.

Users can select the pycharm UI according to your preference, black screen or white screen.

Select and press **Skip Remaining and Set Defaults.** It doesn't matter you select **Next Featured Plugins.**



```
def fib(n):
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a + b
    print()


fib(1000)
```

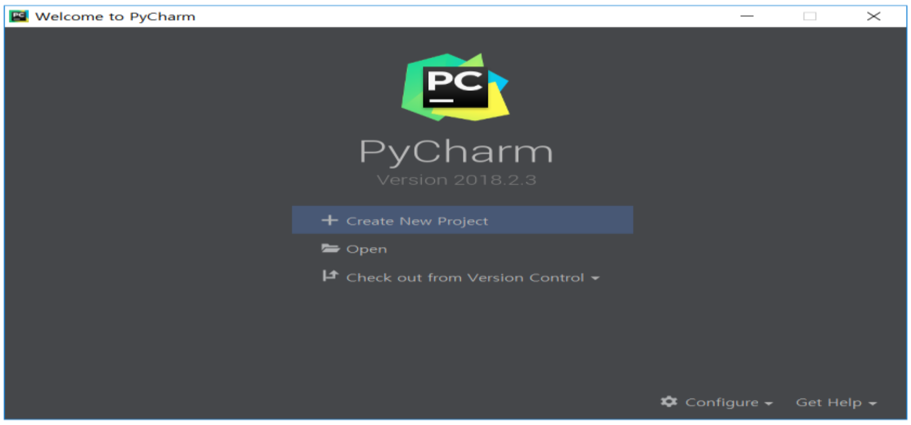
UI theme can be changed later in Settings | Appearance & Behavior | Appearance

2.1.4 Pycharm Execution

Now we will how to draw a ship's calculation table and a hydrostatic curve by running Pycharm.

Download the code file 'shipcurve.py' from the Github site (https://github.com/ARPA-GO/shipcurve) of our project.

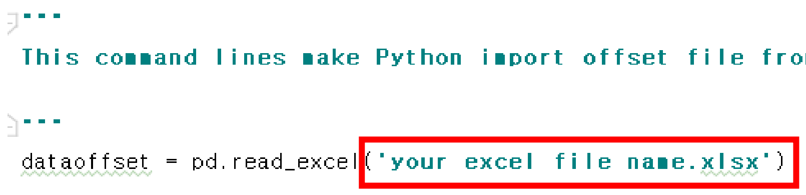
1. Run pycharm. Click [open] to open the downloaded .py file.



2. Import Excel

Import an Excel file which contains offset table of the ship that users want to know. At this point, users must specify the exact location of file.

Python imports Excel file using python itself library called 'Pandas'.



```
'''
This command lines make Python import offset file from Excel.
'''
dataoffset = pd.read_excel('your excel file name.xlsx')
```

3. Before running Pycharm, Add the necessary packages and libraries for coding through the project interpreter. Go to File > Settings > Project: 'project name' > project Interpreter. Users can add the necessary packages by clicking the '+' button in the top right corner.

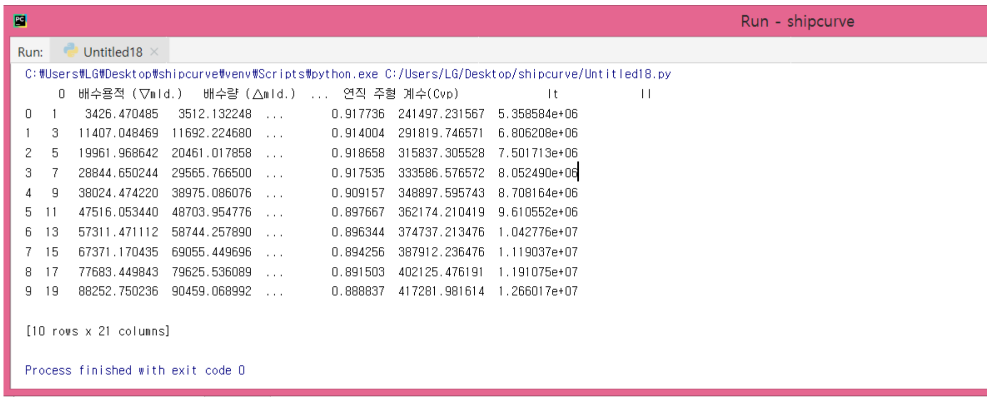
   To create and handle DataFrame, users need a package called Pandas. Numpy is used to manipulate vector data and matrices, and matplotlib is used to visualize data. So users need to install matplotlib, pandas, and numpy to help you analyze your data.

Now, let's get the ship's calculation table and hydrostatic curve through pycharm.

4. Ship Calculation table

To write command lines, users need to use the python console below the window. Run menu doesn't query the dataframe because python exits when execution is over. If you click python console, ipython runs. Place or select the cursor in the code and enter alt+shift+E, the syntax will execute in the console window. But it takes a long time because you have to run by a single command line.

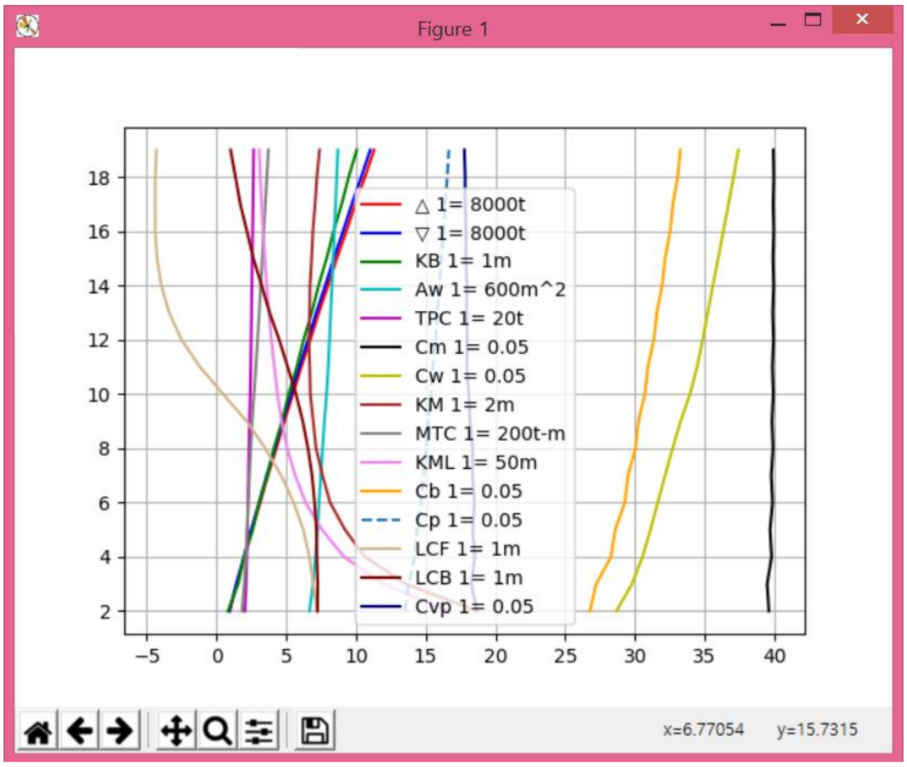
Another way is using the function 'print'. Users can derive a ship calculation table writing the codes, 'print (Calculation_sheet())'.



```
C:₩Users₩LG₩Desktop₩shipcurve₩venv₩Scripts₩python.exe C:/Users/LG/Desktop/shipcurve/Untitled18.py
     0  배수용적 (▽mld.)   배수량 (△mld.)  ...  연직 주형 계수(Cvp)            It            II
0   1    3426.470485   3512.132248  ...         0.917736  241497.231567  5.358584e+06
1   3   11407.048469  11692.224680  ...         0.914004  291819.746571  6.806208e+06
2   5   19961.968642  20461.017858  ...         0.918658  315837.305528  7.501713e+06
3   7   28844.650244  29565.766500  ...         0.917535  333586.576572  8.052490e+06
4   9   38024.474220  38975.086076  ...         0.909157  348897.595743  8.708164e+06
5  11   47516.053440  48703.954776  ...         0.897667  362174.210419  9.610552e+06
6  13   57311.471112  58744.257890  ...         0.896344  374737.213476  1.042776e+07
7  15   67371.170435  69055.449696  ...         0.894256  387912.236476  1.119037e+07
8  17   77683.449843  79625.536089  ...         0.891503  402125.476191  1.191075e+07
9  19   88252.750236  90459.068992  ...         0.888837  417281.981614  1.266017e+07

[10 rows x 21 columns]

Process finished with exit code 0
```

5. Hydrostatic curve

Clicks run **'file_name'** through the Run menu at the top of pycharm. Then, users will able to see the hydrostatic curve is drawn.



2.2 Exe File

If the users want to use ARPA-GO as program form, you can install the execution file we provide. This form is more convenient to get the clear results.

2.2.1 Installation

We recommend users to read README.md written in the Github site of our project ARPA-GO (https://github.com/ARPA-GO/shipcurve). There will be address of our project's public drive (https://drive.google.com/open?id=1QNXKc0-pwct7xsZWkVGubp9pcGPgKCwM).

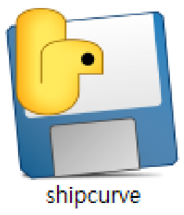
Follow the procedures below to download execution file of ARPA-GO.

Public google drives → GUI form → shipcurve.exe

2.2.2 Execution

① Install **shipcurve** file

(It takes time to open.)



② Import offset file

✓ Users not having offset file

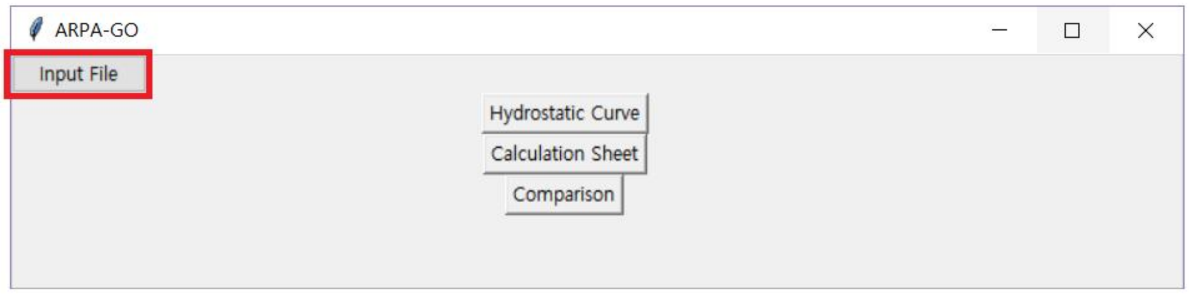
Users who have only drawing plan of ship can easily write offset files by using lisp referring to the appendix 4.1.1 Lisp.

✓ Users having offset file

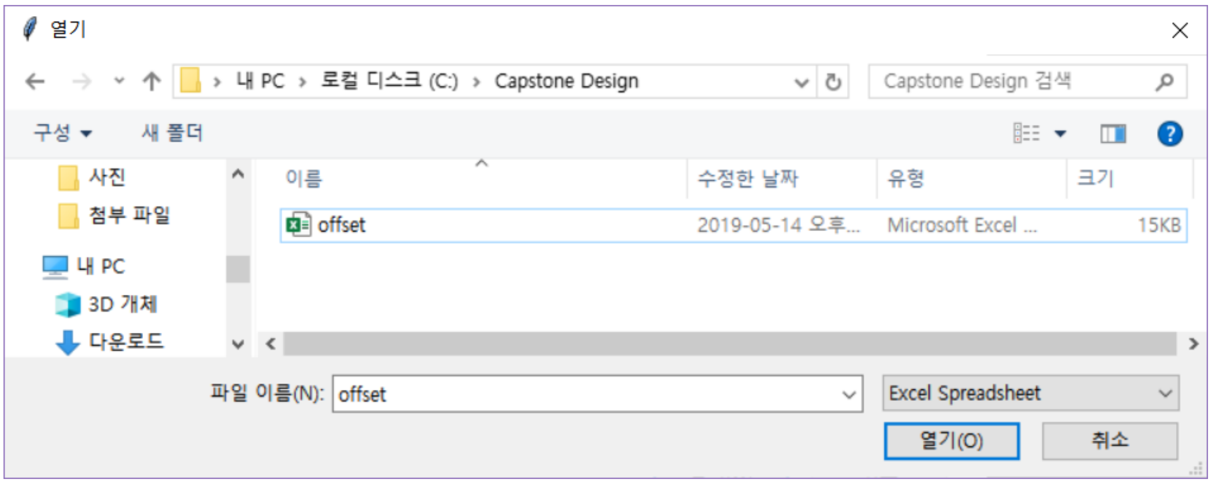
Click the button **'Input File'**.

Select the **offset file** and click **'Open'** button

Users would able to see the file is applied.

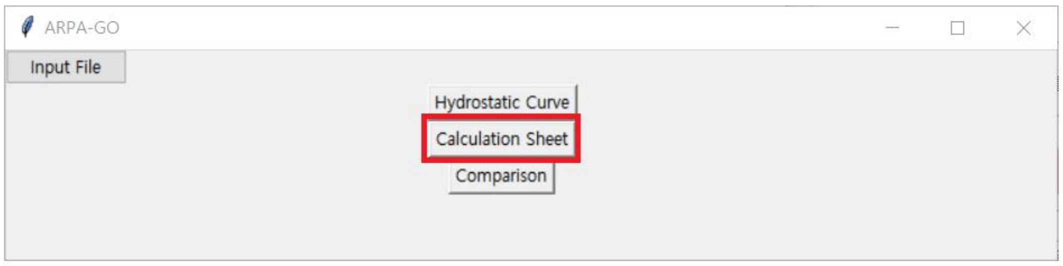


Click 'Input File' button

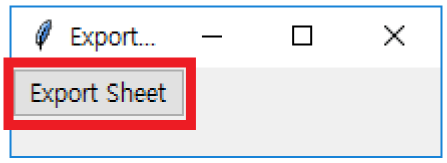


Select offset file and click 'open' button

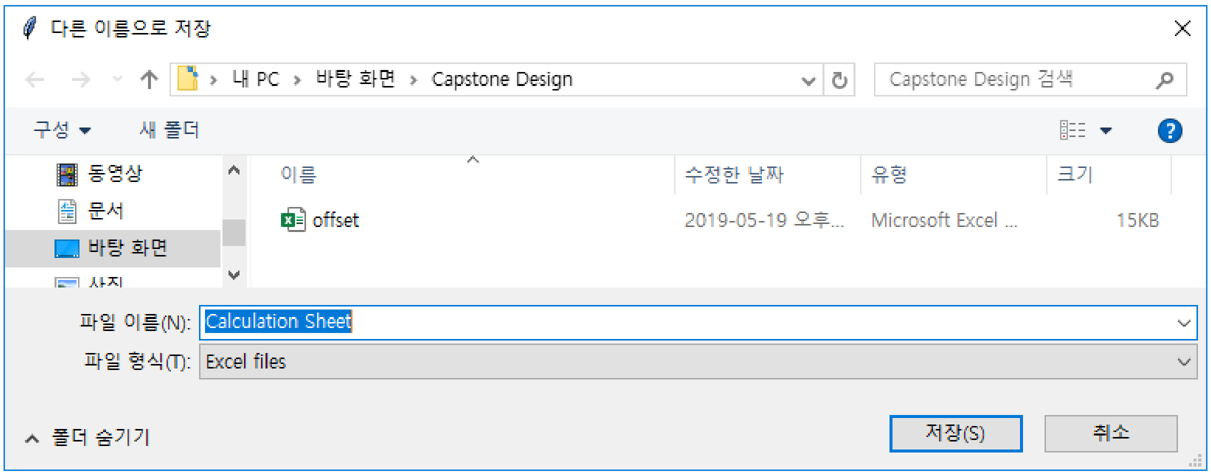
③ Ship Calculation Sheet



Click **'Calculation Sheet'** button to display it.



Click **'Export Sheet'** button.


다른 이름으로 저장

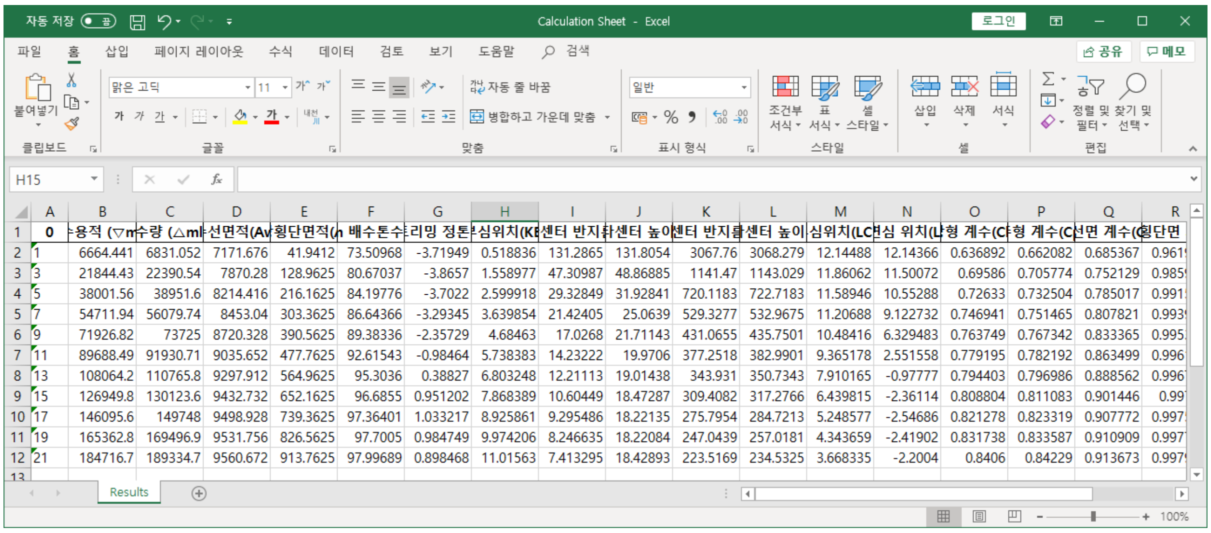
내 PC › 바탕 화면 › Capstone Design

이름	수정한 날짜	유형	크기
offset	2019-05-19 오후...	Microsoft Excel ...	15KB

파일 이름(N): Calculation Sheet

파일 형식(T): Excel files

Then window will come out to determine name and path of file which is to save the **'Calculation sheet'**

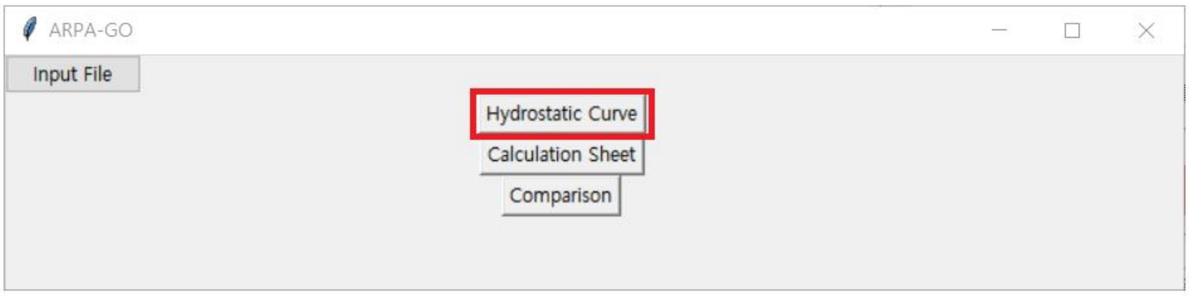

Calculation Sheet - Excel

H15

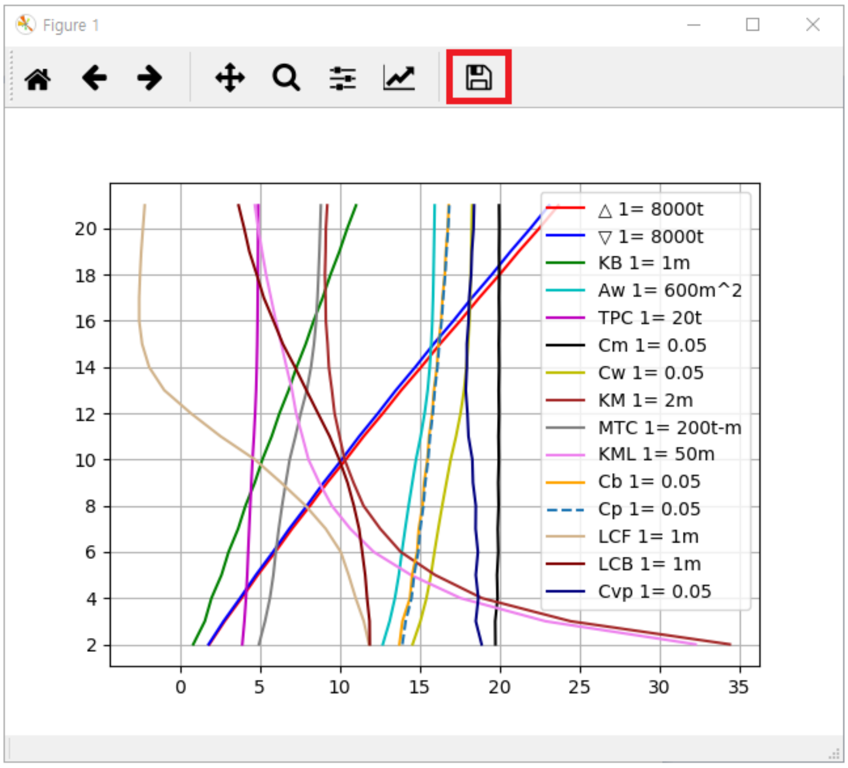
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	0	용적 (▽m	수량 (△ml	선면적(Av	횡단면적(A	배수톤수	리밍 정톤	심위치(KB	센터 반지	센터 높이	센터 반지름	센터 높이	심위치(LC	심 위치(L	형 계수(C	형 계수(C	선면 계수(C	횡단면
2	1	6664.441	6831.052	7171.676	41.9412	73.50968	-3.71949	0.518836	131.2865	131.8054	3067.76	3068.279	12.14488	12.14366	0.636892	0.662082	0.685367	0.961
3	3	21844.43	22390.54	7870.28	128.9625	80.67037	-3.8657	1.558977	47.30987	48.86885	1141.47	1143.029	11.86062	11.50072	0.69586	0.705774	0.752129	0.985
4	5	38001.56	38951.6	8214.416	216.1625	84.19776	-3.7022	2.599918	29.32849	31.92841	720.1183	722.7183	11.58946	10.55288	0.72633	0.732504	0.785017	0.991
5	7	54711.94	56079.74	8453.04	303.3625	86.64366	-3.29345	3.639854	21.42405	25.0639	529.3277	532.9675	11.20688	9.122732	0.746941	0.751465	0.807821	0.993
6	9	71926.82	73725	8720.328	390.5625	89.38336	-2.35729	4.68463	17.0268	21.71143	431.0655	435.7501	10.48416	6.329483	0.763749	0.767342	0.833365	0.995
7	11	89688.49	91930.71	9035.652	477.7625	92.61543	-0.98464	5.738383	14.23222	19.9706	377.2518	382.9901	9.365178	2.551558	0.779195	0.782192	0.863499	0.996
8	13	108064.2	110765.8	9297.912	564.9625	95.3036	0.38827	6.803248	12.21113	19.01438	343.931	350.7343	7.910165	-0.97777	0.794403	0.796986	0.888562	0.996
9	15	126949.8	130123.6	9432.732	652.1625	96.6855	0.951202	7.868389	10.60449	18.47287	309.4082	317.2766	6.439815	-2.36114	0.808804	0.811083	0.901446	0.99
10	17	146095.6	149748	9498.928	739.3625	97.36401	1.033217	8.925861	9.295486	18.22135	275.7954	284.7213	5.248577	-2.54686	0.821278	0.823319	0.907772	0.997
11	19	165362.8	169496.9	9531.756	826.5625	97.7005	0.984749	9.974206	8.246635	18.22084	247.0439	257.0181	4.343659	-2.41902	0.831738	0.833587	0.910909	0.997
12	21	184716.7	189334.7	9560.672	913.7625	97.99689	0.898468	11.01563	7.413295	18.42893	223.5169	234.5325	3.668335	-2.2004	0.8406	0.84229	0.913673	0.997
13																		

As above, Users can see the 'Calculation Sheet' created

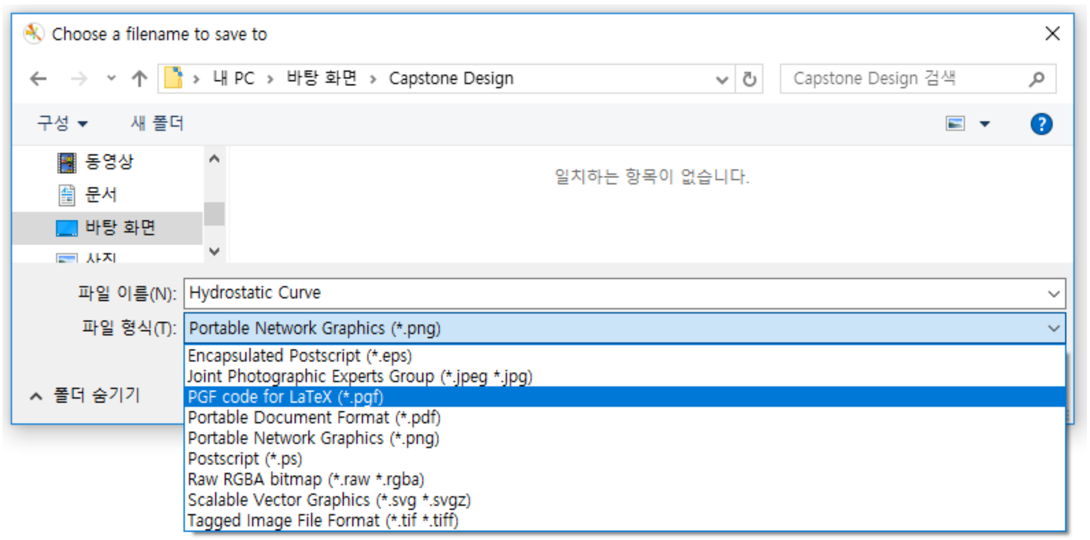
④ Hydrostatic Curve



If you click the Hydrostatic curve button, you can see the figure of hydrostatic curves.



Click the save button to save the curves in the format you prefer.


Choose a filename to save to
내 PC › 바탕 화면 › Capstone Design
Capstone Design 검색
구성
새 폴더
동영상
문서
바탕 화면
사진
일치하는 항목이 없습니다.
폴더 숨기기

파일 이름(N): Hydrostatic Curve

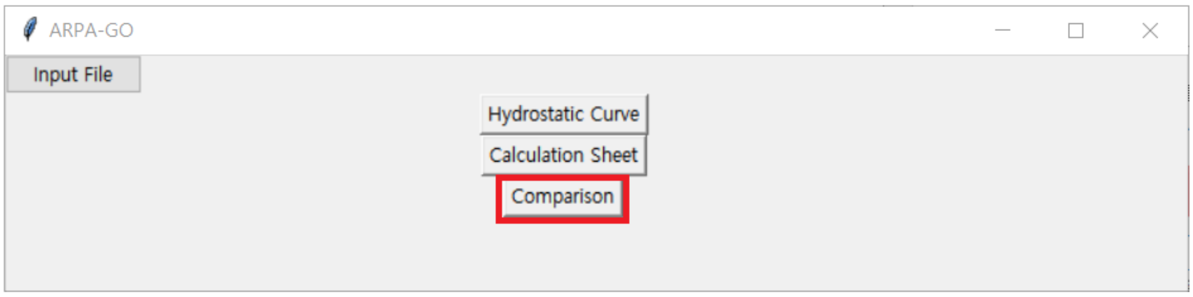
파일 형식(T): Portable Network Graphics (*.png)

Encapsulated Postscript (*.eps)
Joint Photographic Experts Group (*.jpeg *.jpg)
PGF code for LaTeX (*.pgf)
Portable Document Format (*.pdf)
Portable Network Graphics (*.png)
Postscript (*.ps)
Raw RGBA bitmap (*.raw *.rgba)
Scalable Vector Graphics (*.svg *.svgz)
Tagged Image File Format (*.tif *.tiff)

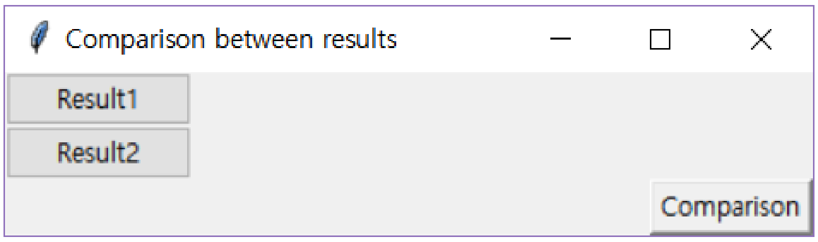
⑤ Comparison of Excel

If users modify drawing plan of ship, the offset file and the calculation sheet changes.

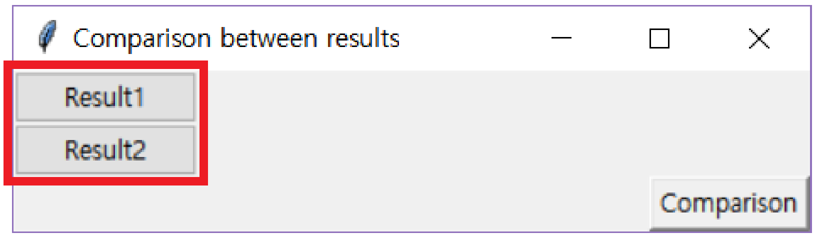
Our project shows the difference between before and after the modification.



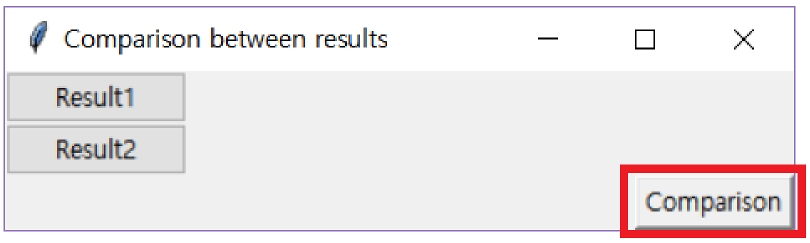
When user clicks 'Comparison' button, it will appear next window.



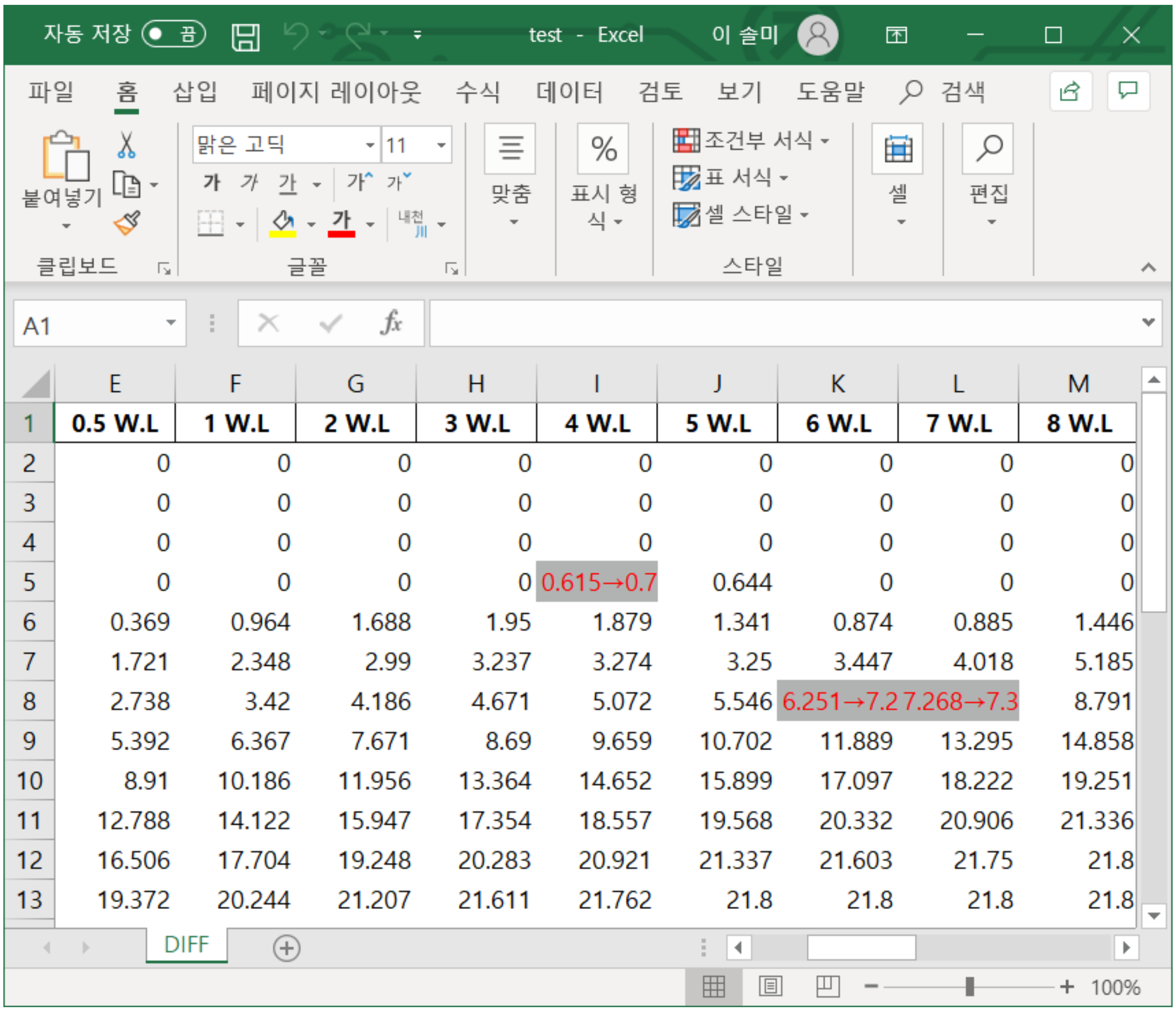
Then user should click 'Result1', 'Result2' button to import Excel file which will be compared.



And finally Click 'Comparison' button and save Excel file. In this file, we marked the values changes.



This is the example of excel file. There are values marked in color. A → B means value A changed to value B



	E	F	G	H	I	J	K	L	M
1	**0.5 W.L**	**1 W.L**	**2 W.L**	**3 W.L**	**4 W.L**	**5 W.L**	**6 W.L**	**7 W.L**	**8 W.L**
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0.615→0.7	0.644	0	0	0
6	0.369	0.964	1.688	1.95	1.879	1.341	0.874	0.885	1.446
7	1.721	2.348	2.99	3.237	3.274	3.25	3.447	4.018	5.185
8	2.738	3.42	4.186	4.671	5.072	5.546	6.251→7.2	7.268→7.3	8.791
9	5.392	6.367	7.671	8.69	9.659	10.702	11.889	13.295	14.858
10	8.91	10.186	11.956	13.364	14.652	15.899	17.097	18.222	19.251
11	12.788	14.122	15.947	17.354	18.557	19.568	20.332	20.906	21.336
12	16.506	17.704	19.248	20.283	20.921	21.337	21.603	21.75	21.8
13	19.372	20.244	21.207	21.611	21.762	21.8	21.8	21.8	21.8

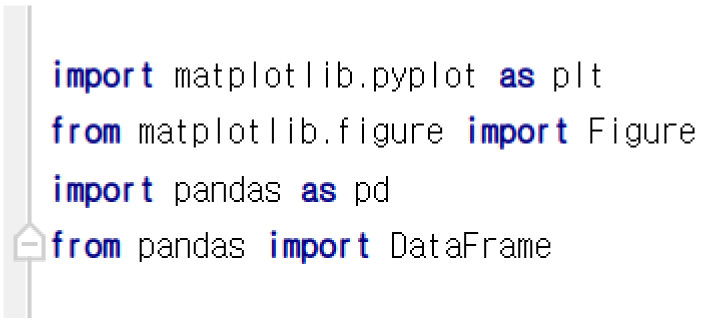
3. code

In this chapter, we will explain the code that constitutes ARPA-GO

3.1 codes

The code refers to writing that uses a computer program as a human-readable programming language.

3.1.1 Python standard library



```
import matplotlib.pyplot as plt
from matplotlib.figure import Figure
import pandas as pd
from pandas import DataFrame
```

There are a lot of libraries that are basically provided by Pycharm. The library used in the project 'ARPA – GO' is 'matplotlib' library to show the hydrostatic Curve and the 'pandas' library to express the Ship Calculation table.

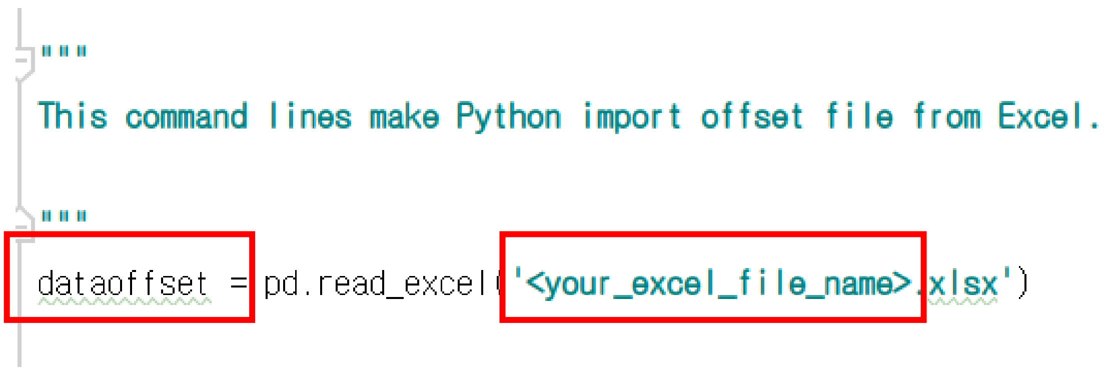
<matplotlib>

Matplotlib is a package that visualizes data in charts or plots in Python.

<pandas>

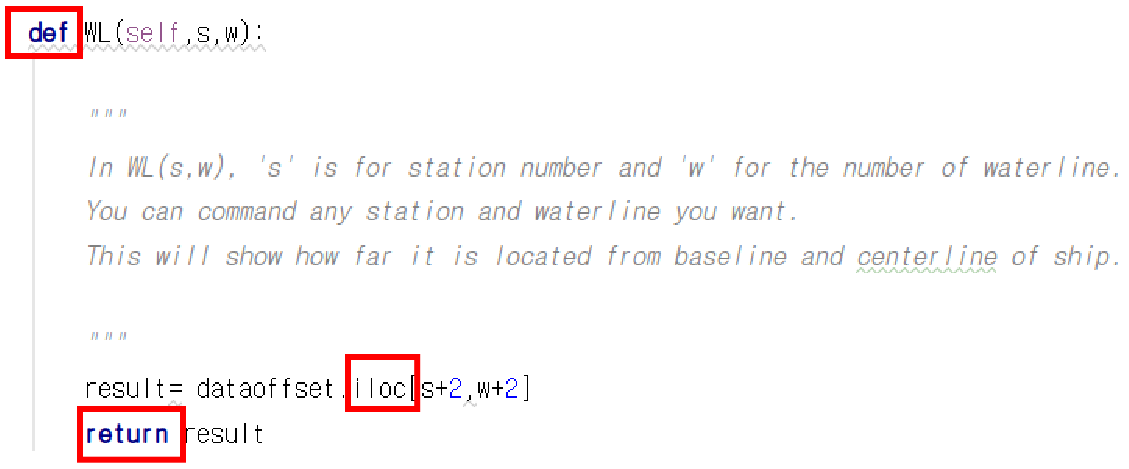
Pandas is a data analysis library used by Python that allows you to create and handle data objects with rows and columns and it is a very convenient tool for handling large volumes of data

3.1.2 Explanation of function



```
"""
This command lines make Python import offset file from Excel.
"""
dataoffset = pd.read_excel('<your_excel_file_name>.xlsx')
```

'pd.read_excel' is a function of standard library 'pandas' and is used to import offset tables of ship in Python language. Pd is abbreviation for pandas, you can make any shortcut of this function. Enter the offset name of the ship you want to know in <your_excel_file_name>, and enter 'dataoffset' in the pycharm. As a result, you can call the offset file.

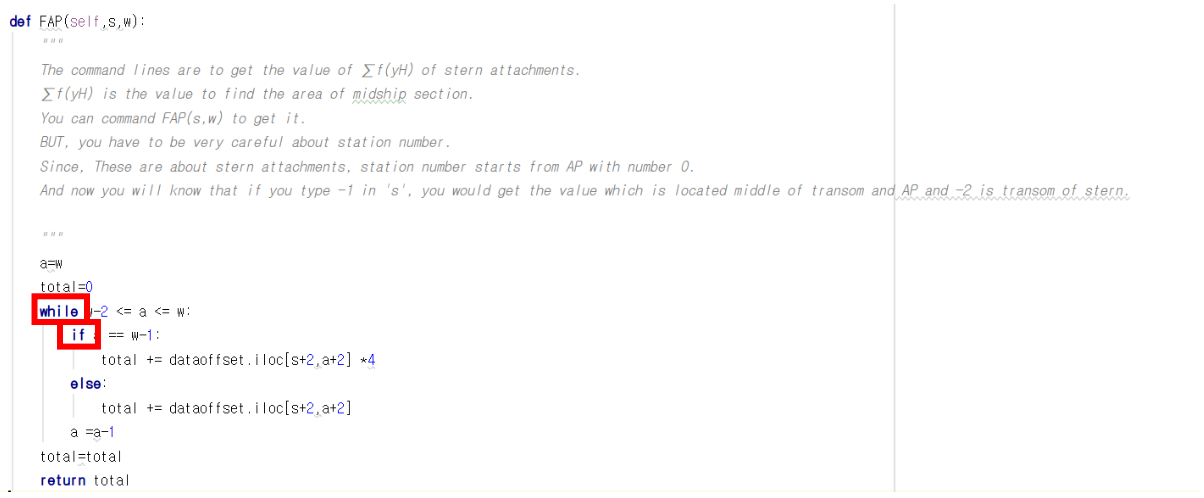


```
def WL(self,s,w):
    """
    In WL(s,w), 's' is for station number and 'w' for the number of waterline.
    You can command any station and waterline you want.
    This will show how far it is located from baseline and centerline of ship.

    """
    result= dataoffset.iloc[s+2,w+2]
    return result
```

'Def' is a command function used to create a function and the function name can be determined by the user. Parameters in parentheses after the function name are entered into this function to obtain the values. After defining the function, you can enter the sentences that you want to perform in the function, just as 'if', 'while', 'for'.

The function 'return' can be used to obtain the final value of this function.

'.iloc' is a command function of 'pandas'. This function extracts the value in the form of rows and columns. Users can enter the coordinates of the Excel file that you want to know. In order to draw a 'Hydrostatic curve', you first need to know the values that make the curve. In order to know these values, you have to know the 'offset' table's value. The command function we defined above is to get the 'offset' value of your desired 'station' and 'waterline'.



```
def FAP(self,s,w):
    """
    The command lines are to get the value of ∑f(yH) of stern attachments.
    ∑f(yH) is the value to find the area of midship section.
    You can command FAP(s,w) to get it.
    BUT, you have to be very careful about station number.
    Since, These are about stern attachments, station number starts from AP with number 0.
    And now you will know that if you type -1 in 's', you would get the value which is located middle of transom and AP and -2 is transom of stern.

    """
    a=w
    total=0
    while w-2 <= a <= w:
        if a == w-1:
            total += dataoffset.iloc[s+2,a+2] *4
        else:
            total += dataoffset.iloc[s+2,a+2]
        a =a-1
    total=total
    return total
```

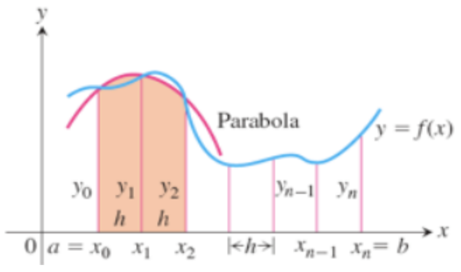
'Simpson' formula applied in the coding is calculated in three groups. This means we divide waterline in three groups. 'While' is a basic function from Python that repeats something within certain condition. Since three 'waterlines' are grouped together to obtain '∑f(yH)', repeat until it ends. The command function 'if' is a function that gives conditions.

심슨의 법칙 *(Simpson's Rule)*

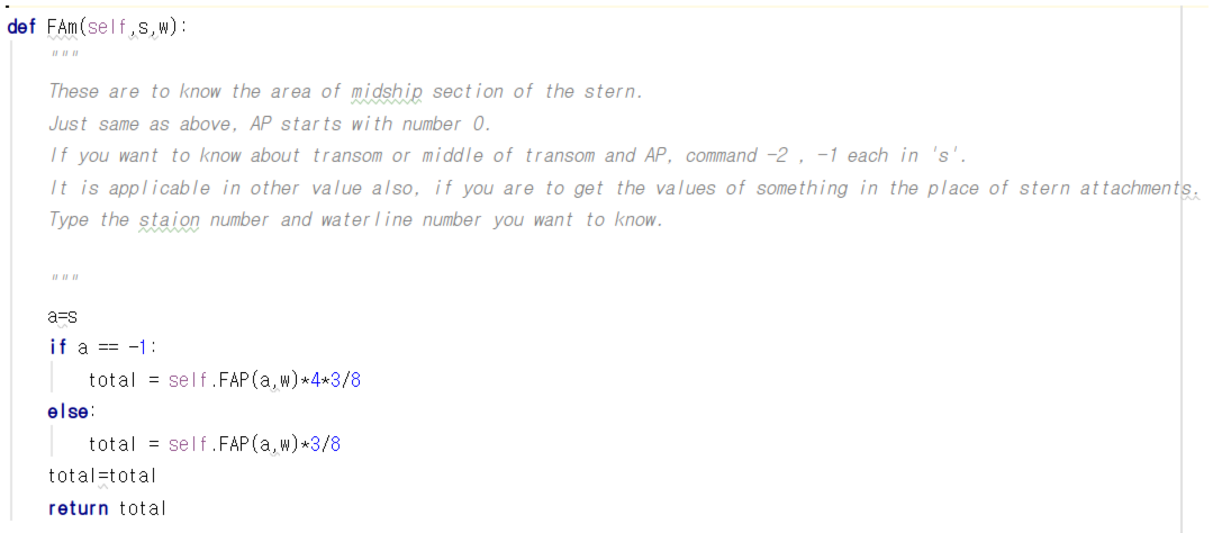
Simpson's rule can be derived by approximating the integrand f (x) (in blue) by the quadratic interpolant P(x) (in red).

$$A_P = \frac{h}{3}(y_0 + 4y_1 + y_2)$$

.

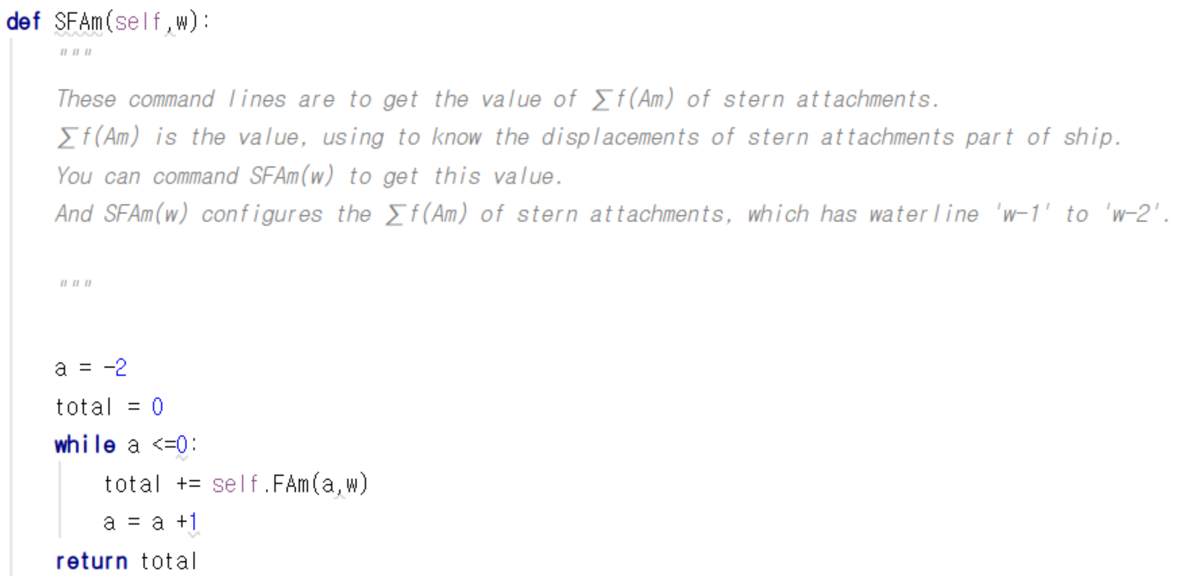


The 'Simpson' formula gives the first value (multiplied by 1), the second value (multiplied by 4) and the third value (multiplied by 1). We used the 'if' command function to each value accordingly.



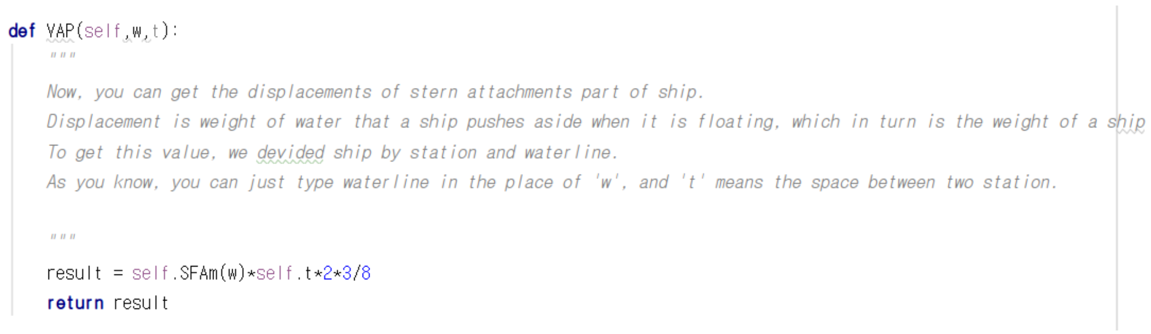
```
def FAm(self,s,w):
    """
    These are to know the area of midship section of the stern.
    Just same as above, AP starts with number 0.
    If you want to know about transom or middle of transom and AP, command -2 , -1 each in 's'.
    It is applicable in other value also, if you are to get the values of something in the place of stern attachments.
    Type the staion number and waterline number you want to know.

    """
    a=s
    if a == -1:
        total = self.FAP(a,w)*4*3/8
    else:
        total = self.FAP(a,w)*3/8
    total=total
    return total
```

The defined command lines above are f(AM) by Stern 'station'. Since we use the Simpson formula, we used 'if' to give different conditions for each 'station'.



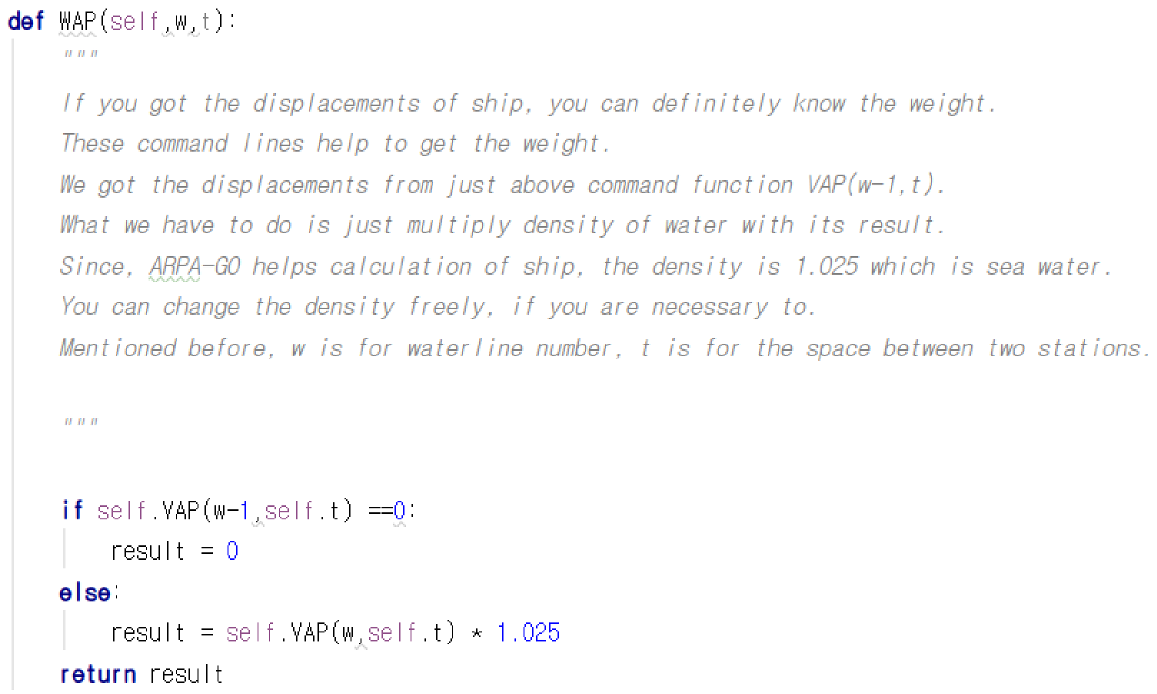
```
def SFAm(self,w):
    """
    These command lines are to get the value of ∑f(Am) of stern attachments.
    ∑f(Am) is the value, using to know the displacements of stern attachments part of ship.
    You can command SFAm(w) to get this value.
    And SFAm(w) configures the ∑f(Am) of stern attachments, which has waterline 'w-1' to 'w-2'.

    """

    a = -2
    total = 0
    while a <=0:
        total += self.FAm(a,w)
        a = a +1
    return total
```

The function 'SFAm' defined above is a ∑f(AM) of the stern and the function of adding f(AM) by the three 'waterlines' obtained above. We made it to add three values by using command repetitive function ‘while’.



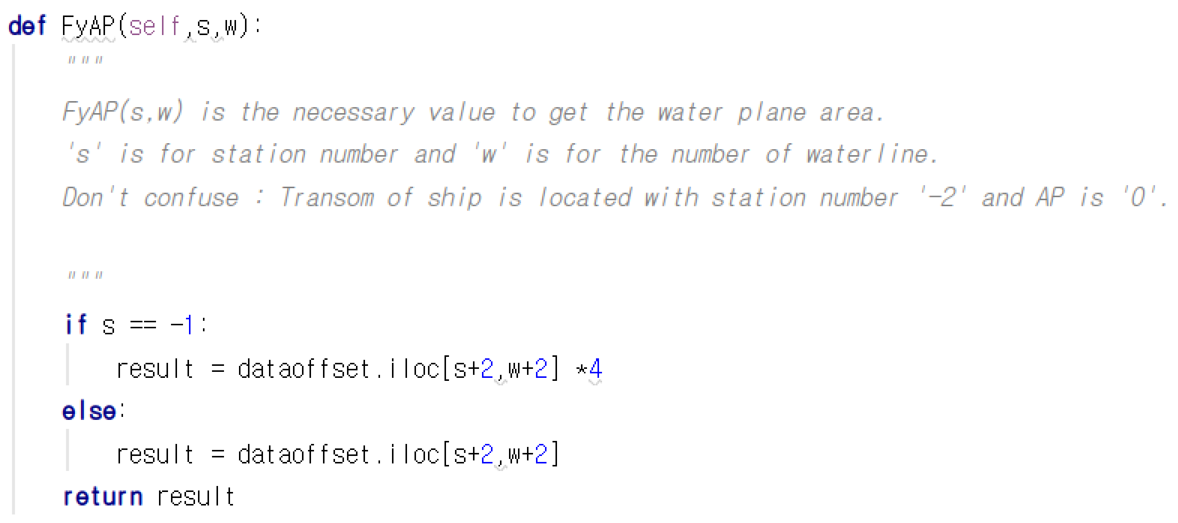
```python
def VAP(self,w,t):
    """
    Now, you can get the displacements of stern attachments part of ship.
    Displacement is weight of water that a ship pushes aside when it is floating, which in turn is the weight of a ship
    To get this value, we devided ship by station and waterline.
    As you know, you can just type waterline in the place of 'w', and 't' means the space between two station.

    """
    result = self.SFAm(w)*self.t*2*3/8
    return result
```

The function defined above is the volume of three parts of certain 'waterline'. The value of f(AM) we got from above is the displacements of stern attachments part of ship. We multiplied the value since our initial principle dimension is half breadth of ship. And also we multiplied by 3/8 in the place of 1/3 to consider about attachments.



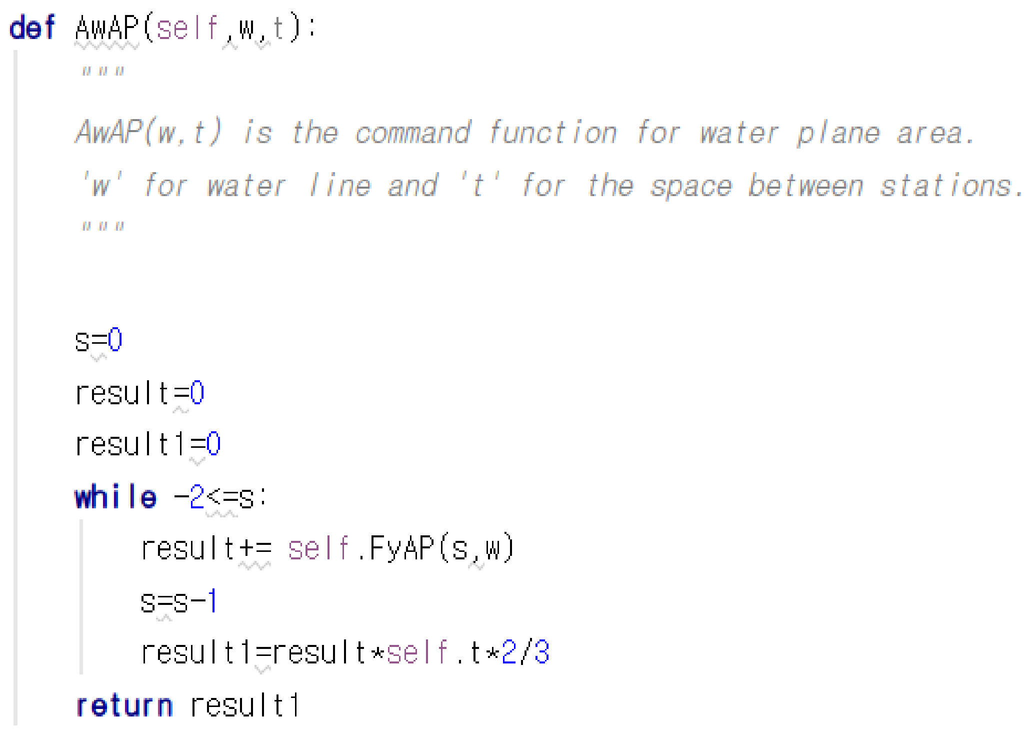
```python
def WAP(self,w,t):
    """
    If you got the displacements of ship, you can definitely know the weight.
    These command lines help to get the weight.
    We got the displacements from just above command function VAP(w-1,t).
    What we have to do is just multiply density of water with its result.
    Since, ARPA-GO helps calculation of ship, the density is 1.025 which is sea water.
    You can change the density freely, if you are necessary to.
    Mentioned before, w is for waterline number, t is for the space between two stations.

    """

    if self.VAP(w-1,self.t) ==0:
        result = 0
    else:
        result = self.VAP(w,self.t) * 1.025
    return result
```

The function defined above is the weight of three parts of certain 'waterline'. We multiplied the density of water with displacements. Since ARPA-GO helps calculation of ship, the density is 1.025(ton/m^3) which is sea water.



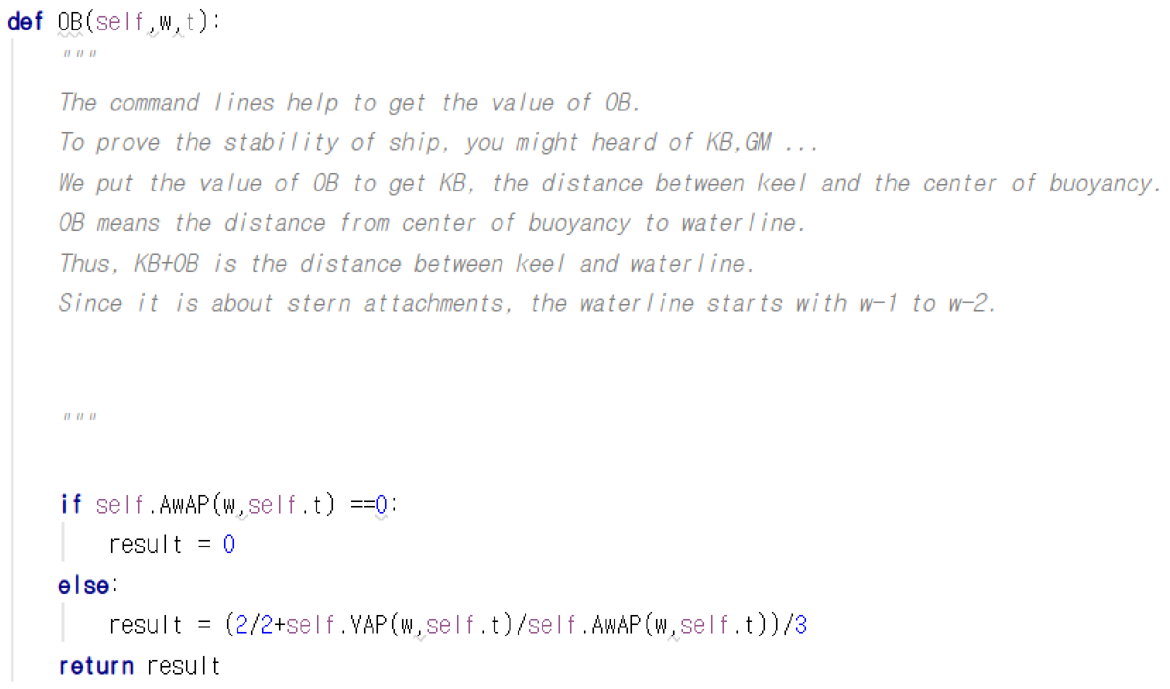
```
def FyAP(self,s,w):
    """
    FyAP(s,w) is the necessary value to get the water plane area.
    's' is for station number and 'w' is for the number of waterline.
    Don't confuse : Transom of ship is located with station number '-2' and AP is '0'.

    """
    if s == -1:
        result = dataoffset.iloc[s+2,w+2] *4
    else:
        result = dataoffset.iloc[s+2,w+2]
    return result
```

The function defined above is a function to obtain f(y) of Stern. In the previous function, 'waterline' is divided into three parts in one station whereas this function is divided into three 'station' in one 'waterline'. Since it uses the 'simpson' formula again, the command function 'if' is used to give different conditions.



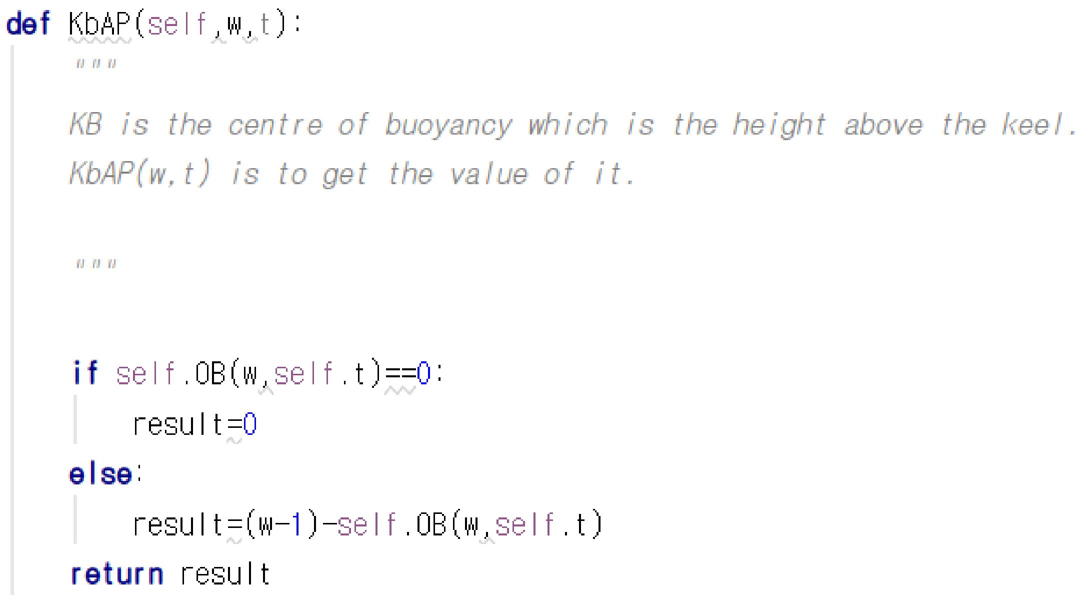
```
def AwAP(self,w,t):
    """
    AwAP(w,t) is the command function for water plane area.
    'w' for water line and 't' for the space between stations.
    """


    s=0
    result=0
    result1=0
    while -2<=s:
        result+= self.FyAP(s,w)
        s=s-1
        result1=result*self.t*2/3
    return result1
```

The command defined above is a function to obtain the 'Aw' of the stern. It adds the f(y) of the 'station' obtained just before. We multiplied the values by 2 and 1/3 because of half breadth of ship and Simpson's rule. We use command function 'while' to add three values in onetime since its results are combined by 3 stations.



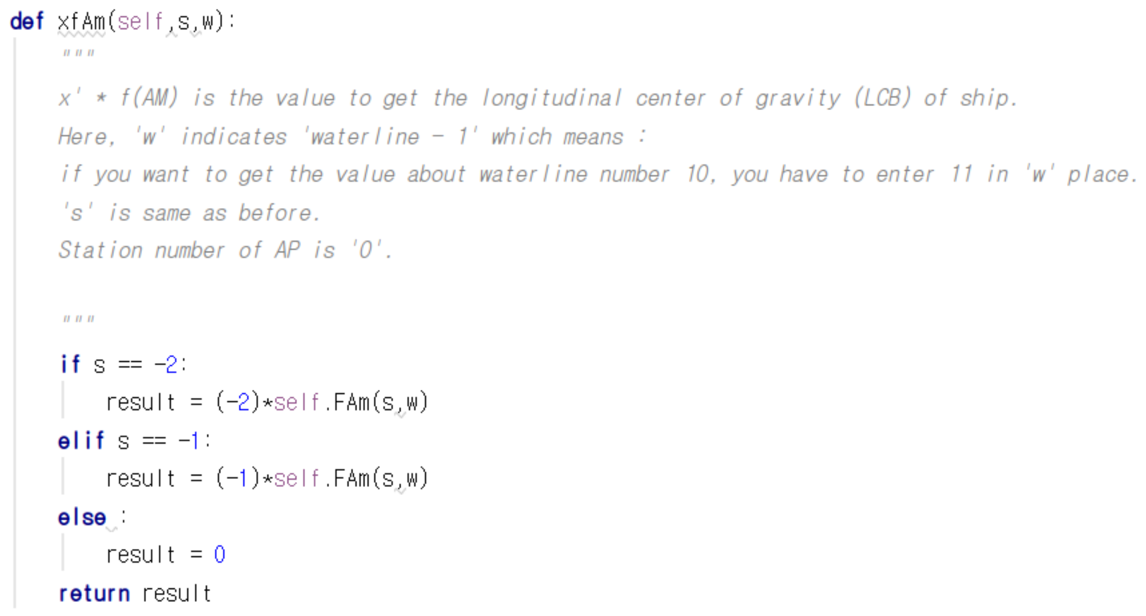
```
def OB(self,w,t):
    """
    The command lines help to get the value of OB.
    To prove the stability of ship, you might heard of KB,GM ...
    We put the value of OB to get KB, the distance between keel and the center of buoyancy.
    OB means the distance from center of buoyancy to waterline.
    Thus, KB+OB is the distance between keel and waterline.
    Since it is about stern attachments, the waterline starts with w-1 to w-2.


    """

    if self.AwAP(w,self.t) ==0:
        result = 0
    else:
        result = (2/2+self.VAP(w,self.t)/self.AwAP(w,self.t))/3
    return result
```

The function defined above is the distance between keel and the center of buoyancy.



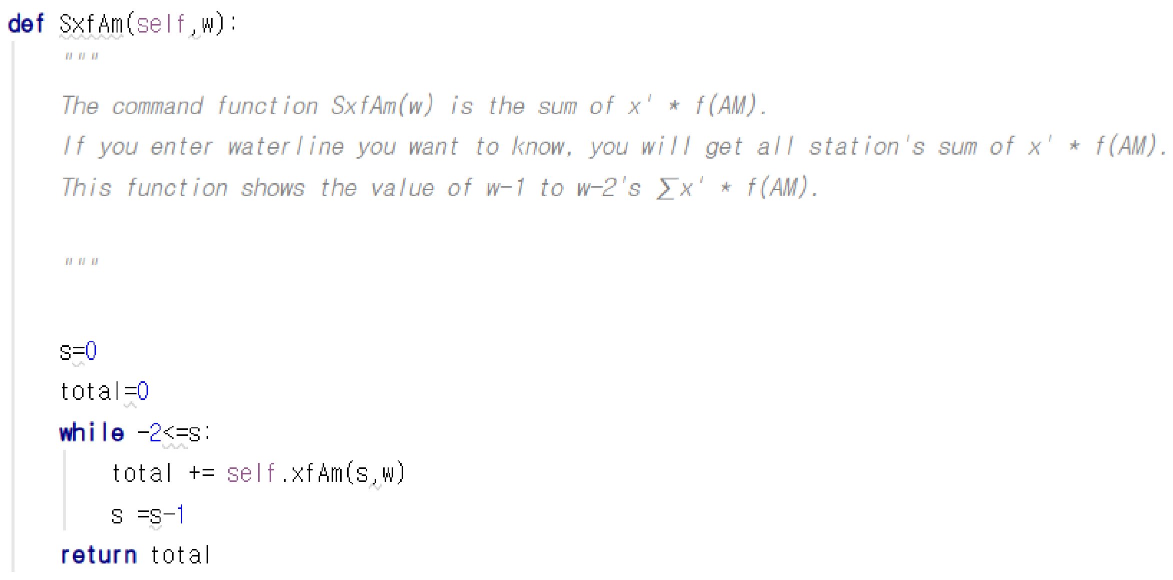
```
def KbAP(self,w,t):
    """
    KB is the centre of buoyancy which is the height above the keel.
    KbAP(w,t) is to get the value of it.

    """

    if self.OB(w,self.t)==0:
        result=0
    else:
        result=(w-1)-self.OB(w,self.t)
    return result
```

The function defined above is kb of stern attachments of ship.



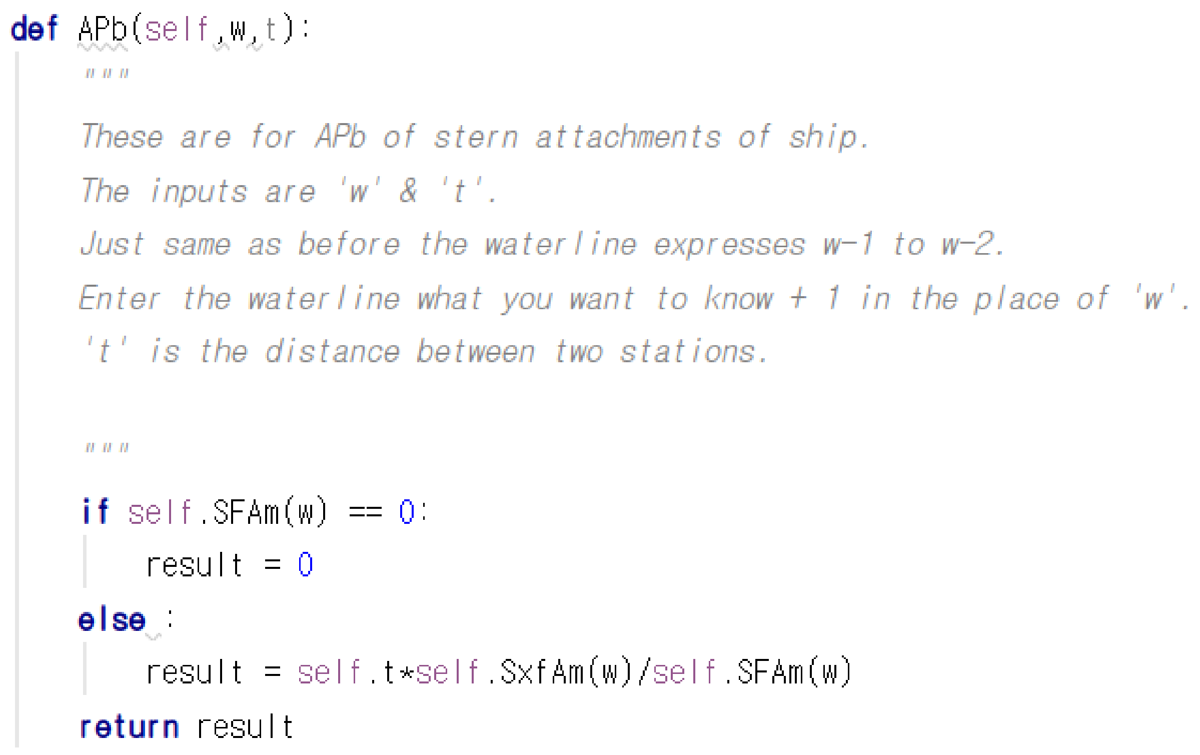
```
def xfAm(self,s,w):
    """
    x' * f(AM) is the value to get the longitudinal center of gravity (LCB) of ship.
    Here, 'w' indicates 'waterline - 1' which means :
    if you want to get the value about waterline number 10, you have to enter 11 in 'w' place.
    's' is same as before.
    Station number of AP is '0'.

    """
    if s == -2:
        result = (-2)*self.FAm(s,w)
    elif s == -1:
        result = (-1)*self.FAm(s,w)
    else :
        result = 0
    return result
```

Let us divide the stations of ship one by one and call its value x'. This command function is to get x'*f(AM). If the station is in a different position, the value of 'x' also changes. That is why we used command function 'if' to give different conditions.



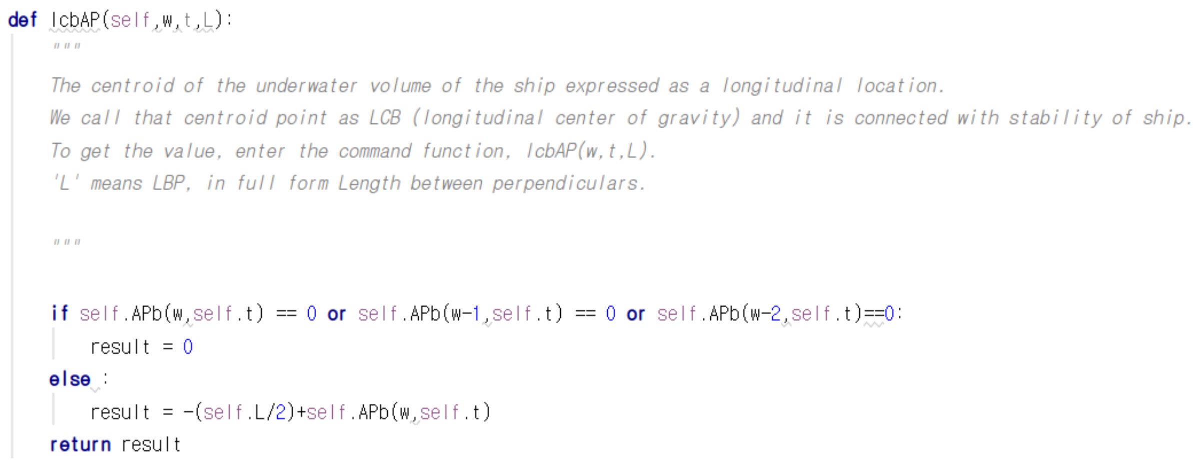
```
def SxfAm(self,w):
    """
    The command function SxfAm(w) is the sum of x' * f(AM).
    If you enter waterline you want to know, you will get all station's sum of x' * f(AM).
    This function shows the value of w-1 to w-2's ∑x' * f(AM).

    """

    s=0
    total=0
    while -2<=s:
        total += self.xfAm(s,w)
        s =s-1
    return total
```

The function above is the sum of the x' * f(AM) of stern. Its result is not entire value of ship, it is just three waterlines combined. We used command function 'while' to do this.



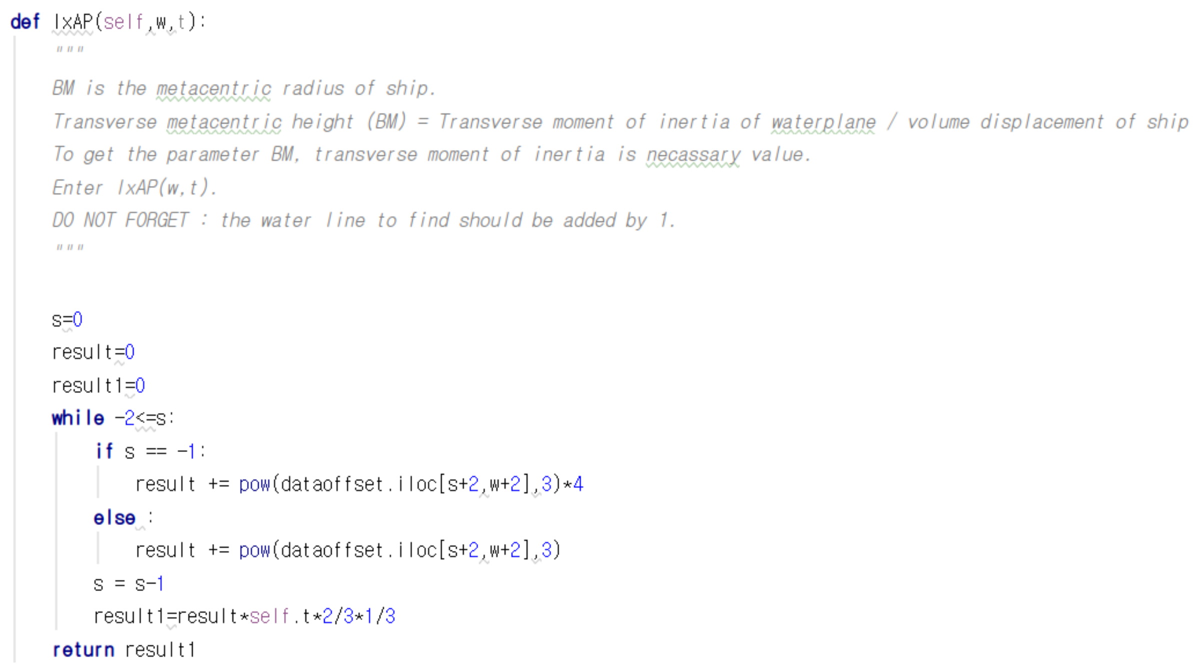
```python
def APb(self,w,t):
    """

    These are for APb of stern attachments of ship.
    The inputs are 'w' & 't'.
    Just same as before the waterline expresses w-1 to w-2.
    Enter the waterline what you want to know + 1 in the place of 'w'.
    't' is the distance between two stations.


    """
    if self.SFAm(w) == 0:
        result = 0
    else :
        result = self.t*self.SxfAm(w)/self.SFAm(w)
    return result
```

The function defined above is the center of gravity of the stern. The center of gravity of the stern can obtain by dividing the previous value '∑x' * f(AM)' by '∑f(AM).



```python
def lcbAP(self,w,t,L):
    """

    The centroid of the underwater volume of the ship expressed as a longitudinal location.
    We call that centroid point as LCB (longitudinal center of gravity) and it is connected with stability of ship.
    To get the value, enter the command function, lcbAP(w,t,L).
    'L' means LBP, in full form Length between perpendiculars.

    """

    if self.APb(w,self.t) == 0 or self.APb(w-1,self.t) == 0 or self.APb(w-2,self.t)==0:
        result = 0
    else :
        result = -(self.L/2)+self.APb(w,self.t)
    return result
```

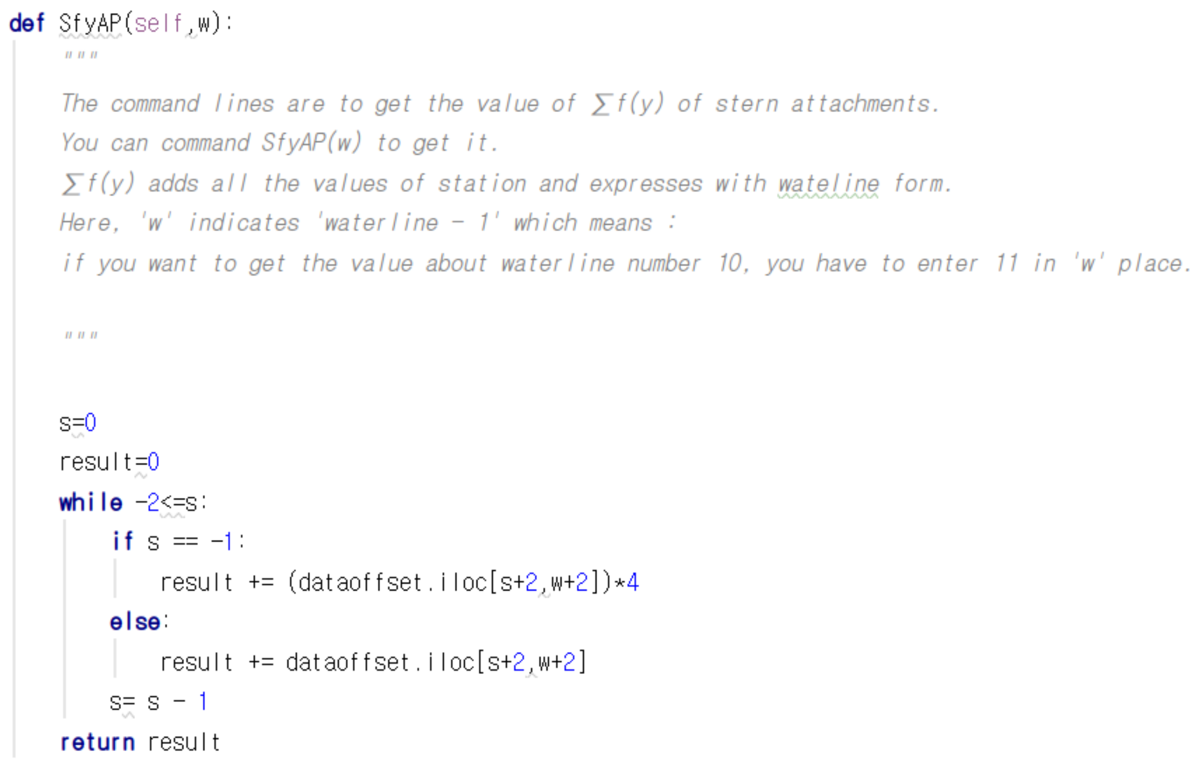
The command lines defined above is the 'LCB' of the ship's three waterlines. Divide the main dimension LBP in half and add the center of gravity of the stern obtained earlier in it to get LCB.



```
def IxAP(self,w,t):
    """
    BM is the metacentric radius of ship.
    Transverse metacentric height (BM) = Transverse moment of inertia of waterplane / volume displacement of ship
    To get the parameter BM, transverse moment of inertia is necassary value.
    Enter IxAP(w,t).
    DO NOT FORGET : the water line to find should be added by 1.
    """

    s=0
    result=0
    result1=0
    while -2<=s:
        if s == -1:
            result += pow(dataoffset.iloc[s+2,w+2],3)*4
        else :
            result += pow(dataoffset.iloc[s+2,w+2],3)
        s = s-1
        result1=result*self.t*2/3*1/3
    return result1
```

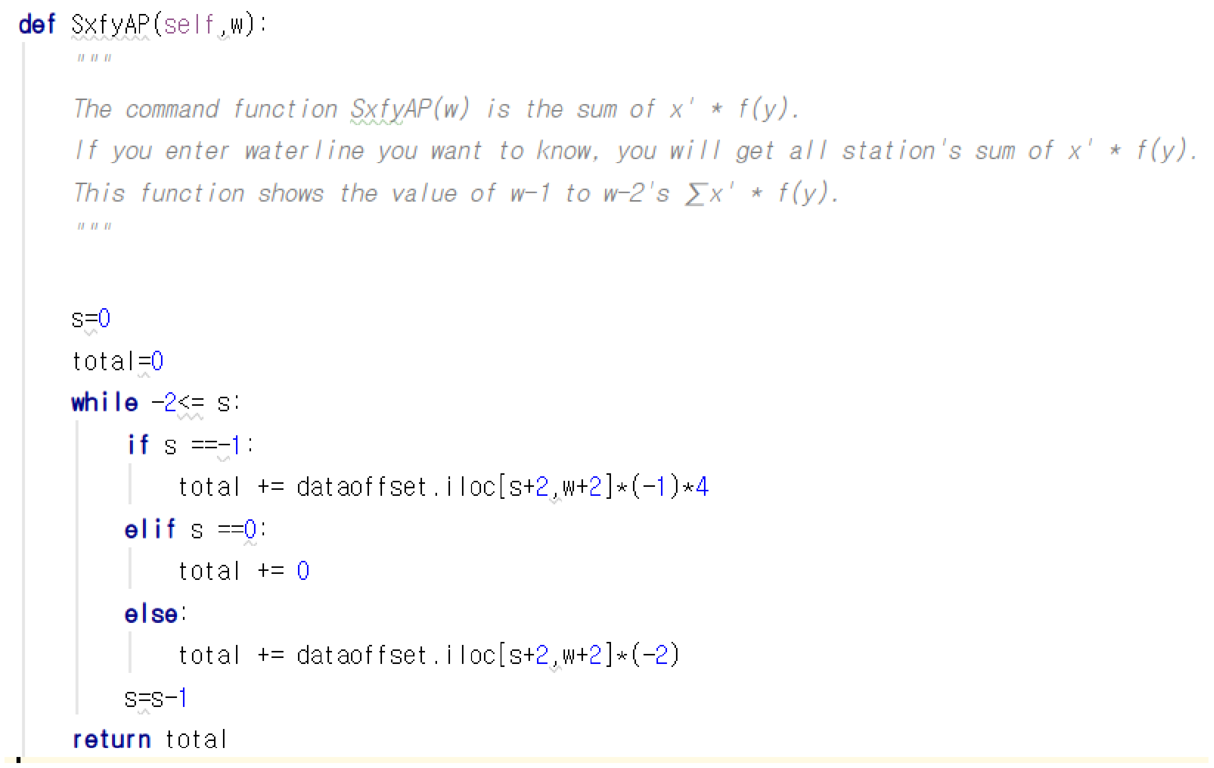
The function defined above is to obtain Ix, which is the secondary moment of the stern.

Using the 'simpson' formula, command function 'if' is used to give different conditions for each location, and also 'while' is used to obtain Ix by three waterlines combined.



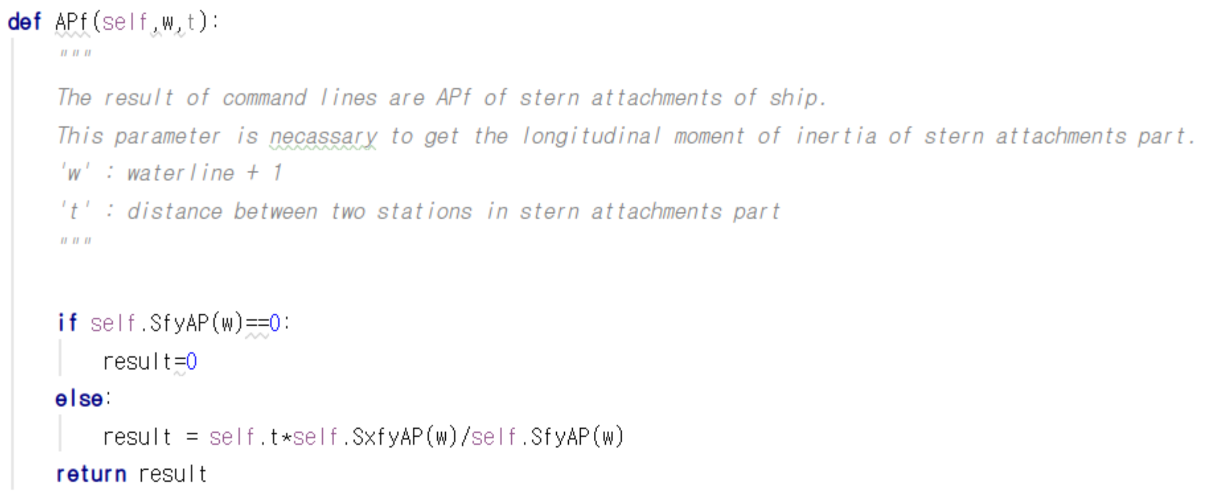
```
def SfyAP(self,w):
    """
    The command lines are to get the value of ∑f(y) of stern attachments.
    You can command SfyAP(w) to get it.
    ∑f(y) adds all the values of station and expresses with wateline form.
    Here, 'w' indicates 'waterline - 1' which means :
    if you want to get the value about waterline number 10, you have to enter 11 in 'w' place.

    """

    s=0
    result=0
    while -2<=s:
        if s == -1:
            result += (dataoffset.iloc[s+2,w+2])*4
        else:
            result += dataoffset.iloc[s+2,w+2]
        s= s - 1
    return result
```

The function defined above is ∑f(y) of three 'stations' in one 'waterline' of the stern. Using the 'simpson' formula, command function 'if' is used to give different conditions for each location, and also 'while' is used to obtain ∑f(y) by three stations combined.



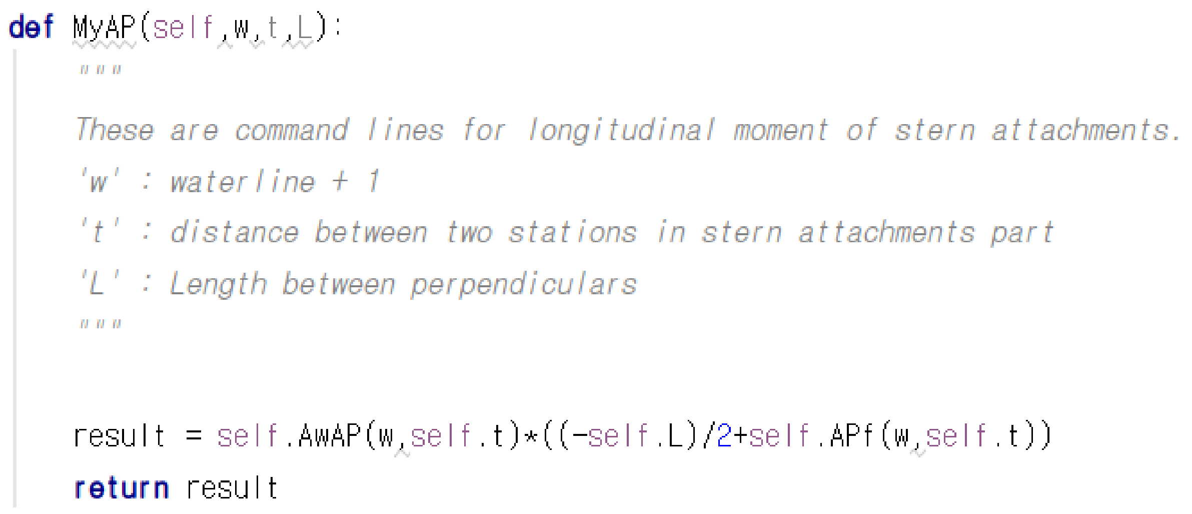
```python
def SxfyAP(self,w):
    """
    The command function SxfyAP(w) is the sum of x' * f(y).
    If you enter waterline you want to know, you will get all station's sum of x' * f(y).
    This function shows the value of w-1 to w-2's ∑x' * f(y).
    """

    s=0
    total=0
    while -2<= s:
        if s ==-1:
            total += dataoffset.iloc[s+2,w+2]*(-1)*4
        elif s ==0:
            total += 0
        else:
            total += dataoffset.iloc[s+2,w+2]*(-2)
        s=s-1
    return total
```

The commands defined above is ∑x' * f(y) of the stern. Since the 'Simpson formula' is used and also the value 'x' is changed for each station, command function 'if' was used to give different conditions for each station and also 'while' is used to obtain three stations combined ∑x' * f(y).



```python
def APf(self,w,t):
    """
    The result of command lines are APf of stern attachments of ship.
    This parameter  is necassary to get the longitudinal moment of inertia of stern attachments part.
    'w' : waterline + 1
    't' : distance between two stations in stern attachments part
    """

    if self.SfyAP(w)==0:
        result=0
    else:
        result = self.t*self.SxfyAP(w)/self.SfyAP(w)
    return result
```

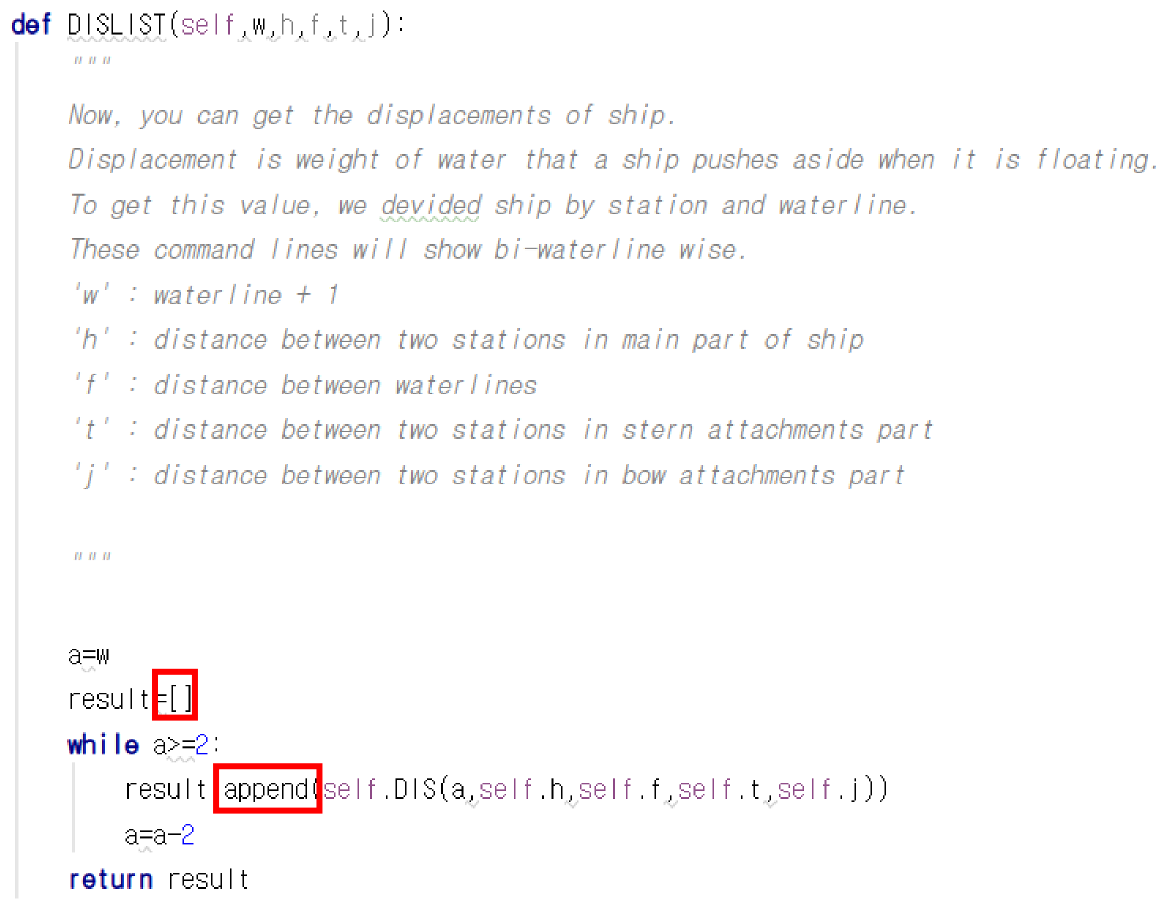
The function defined above is the position of the vertical direction of the center of gravity of the stern. It is the value of ' ∑x' * f(y)' divided by '∑f(y)'.



```python
def MyAP(self,w,t,L):
    """
    These are command lines for longitudinal moment of stern attachments.
    'w' : waterline + 1
    't' : distance between two stations in stern attachments part
    'L' : Length between perpendiculars
    """

    result = self.AwAP(w,self.t)*((-self.L)/2+self.APf(w,self.t))
    return result
```

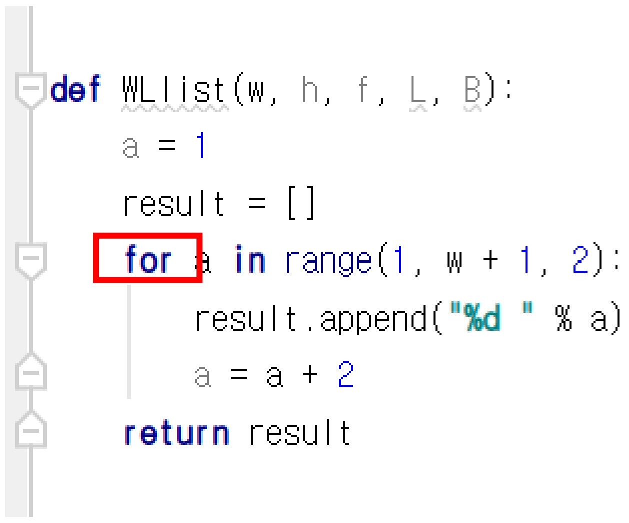
These are command lines for longitudinal moment of the stern attachments.

Our ARPA-GO progresses the calculation of stern / bow attachments and main part of ship according to this. Then, we combine all these three parts of ship and calculate the values which will be ship calculation table and hydrostatic curves.



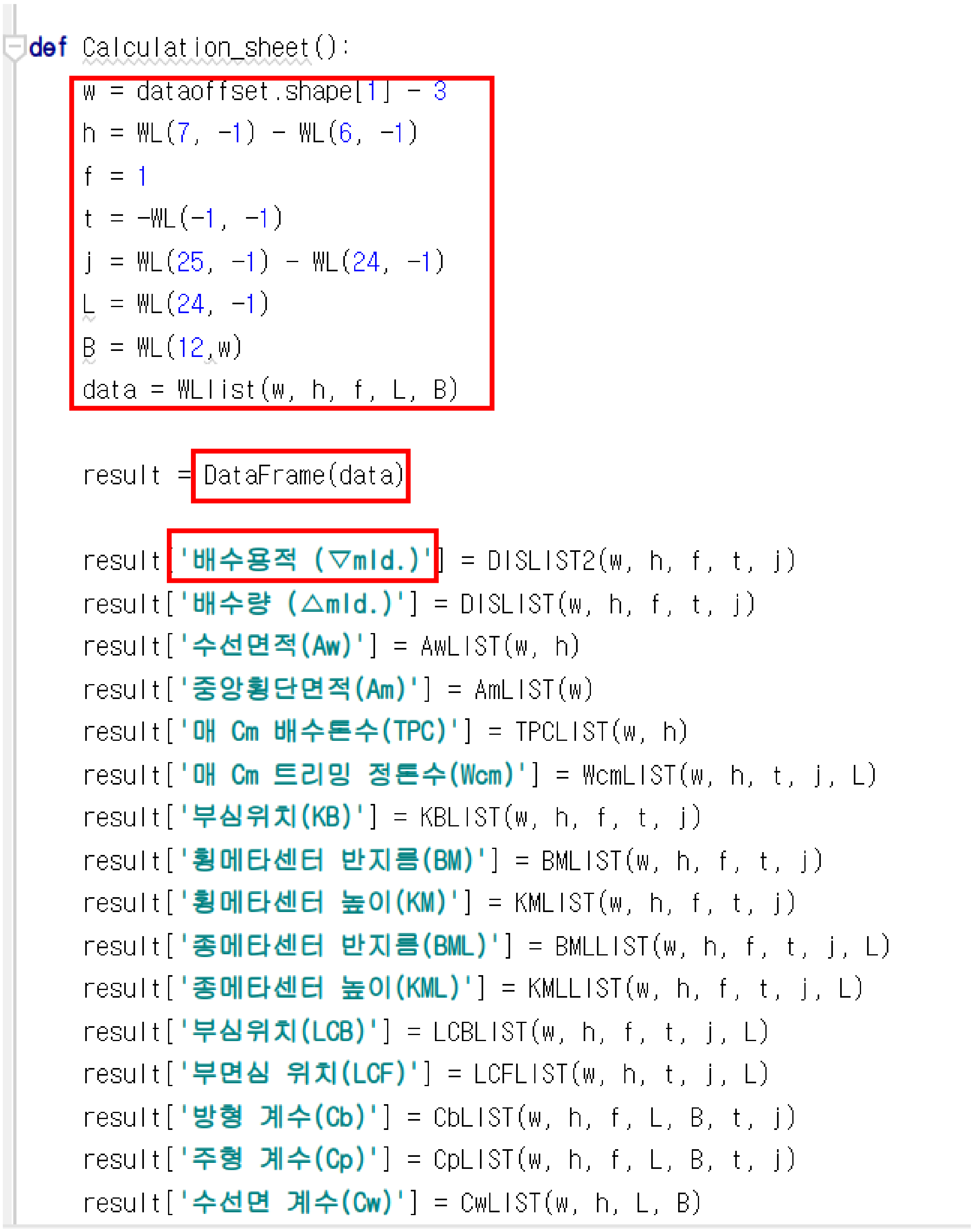
```python
def DISLIST(self,w,h,f,t,j):
    """
    Now, you can get the displacements of ship.
    Displacement is weight of water that a ship pushes aside when it is floating.
    To get this value, we devided ship by station and waterline.
    These command lines will show bi-waterline wise.
    'w' : waterline + 1
    'h' : distance between two stations in main part of ship
    'f' : distance between waterlines
    't' : distance between two stations in stern attachments part
    'j' : distance between two stations in bow attachments part

    """

    a=w
    result=[]
    while a>=2:
        result.append(self.DIS(a,self.h,self.f,self.t,self.j))
        a=a-2
    return result
```

From the result of this command lines above can see the total displacement of each waterline. The command function 'append' has changed each value into the form of a list and must be defined as '[ ]' before creating a list.

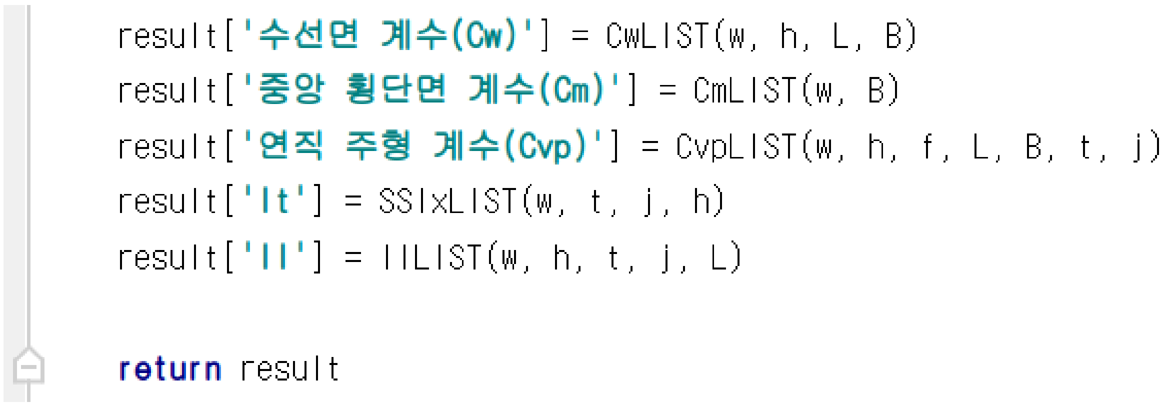


```
def WLlist(w, h, f, L, B):
    a = 1
    result = []
    for a in range(1, w + 1, 2):
        result.append("%d " % a)
        a = a + 2
    return result
```

The command lines defined above use command function 'for' to show the 'waterline' number of in the left side of the ship calculation table.



```
def Calculation_sheet():
    w = dataoffset.shape[1] - 3
    h = WL(7, -1) - WL(6, -1)
    f = 1
    t = -WL(-1, -1)
    j = WL(25, -1) - WL(24, -1)
    L = WL(24, -1)
    B = WL(12,w)
    data = WLlist(w, h, f, L, B)

    result = DataFrame(data)

    result['배수용적 (▽mld.)'] = DISLIST2(w, h, f, t, j)
    result['배수량 (△mld.)'] = DISLIST(w, h, f, t, j)
    result['수선면적(Aw)'] = AwLIST(w, h)
    result['중앙횡단면적(Am)'] = AmLIST(w)
    result['매 Cm 배수톤수(TPC)'] = TPCLIST(w, h)
    result['매 Cm 트리밍 정톤수(Wcm)'] = WcmLIST(w, h, t, j, L)
    result['부심위치(KB)'] = KBLIST(w, h, f, t, j)
    result['횡메타센터 반지름(BM)'] = BMLIST(w, h, f, t, j)
    result['횡메타센터 높이(KM)'] = KMLIST(w, h, f, t, j)
    result['종메타센터 반지름(BML)'] = BMLLIST(w, h, f, t, j, L)
    result['종메타센터 높이(KML)'] = KMLLIST(w, h, f, t, j, L)
    result['부심위치(LCB)'] = LCBLIST(w, h, f, t, j, L)
    result['부면심 위치(LCF)'] = LCFLIST(w, h, t, j, L)
    result['방형 계수(Cb)'] = CbLIST(w, h, f, L, B, t, j)
    result['주형 계수(Cp)'] = CpLIST(w, h, f, L, B, t, j)
    result['수선면 계수(Cw)'] = CwLIST(w, h, L, B)
```



```
    result['수선면 계수(Cw)'] = CwLIST(w, h, L, B)
    result['중앙 횡단면 계수(Cm)'] = CmLIST(w, B)
    result['연직 주형 계수(Cvp)'] = CvpLIST(w, h, f, L, B, t, j)
    result['It'] = SSIxLIST(w, t, j, h)
    result['Il'] = IlLIST(w, h, t, j, L)

    return result
```

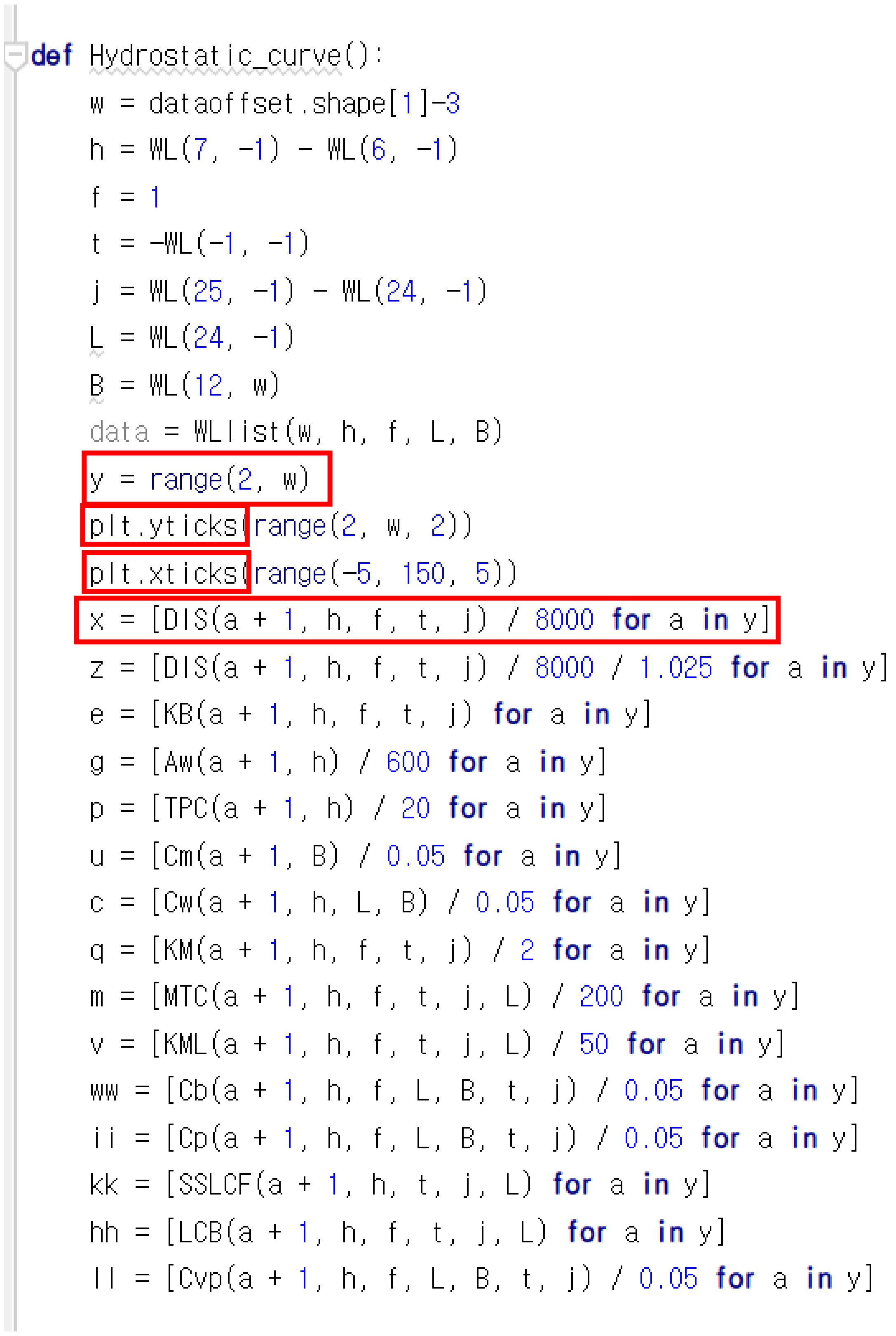
The command lines above collect the results calculated so far and show them in a ship calculation table. Create a table using command function 'dataframe', a function built from 'Pandas', and write down [‵group name′] to see what each value means. The values we got before had to enter the main dimension. But the most of users use our ARPA-GO to obtain the ship calculation table and the hydrostatic curve, the offset table automatically receives each main dimension and displays the ship calculation table if you enter the function name

'w' means the last waterline number using the function '.shape[1]' from library 'pandas'

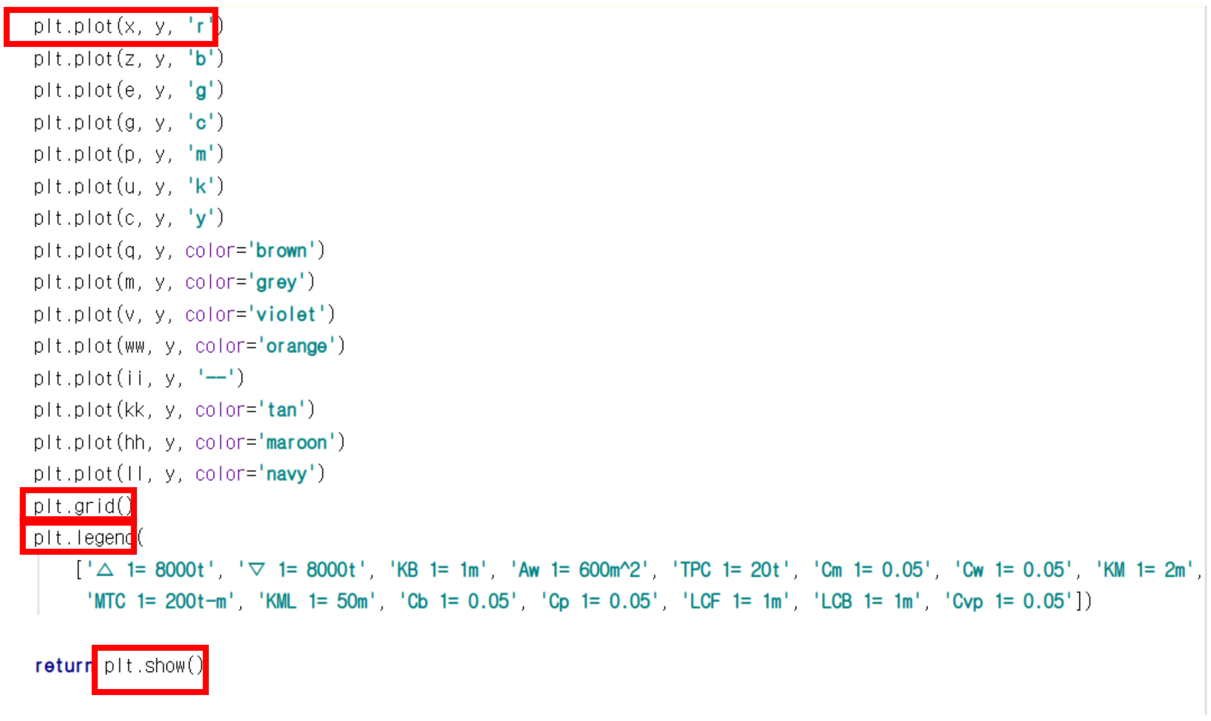
'h' is the distance between two stations of the main part and 't' for distance between two stations of the stern. 'j' is the distance between two stations of the bow. These values are also accessible from the offset table.

'L' means the LBP of the ship. We import the distance till stern attachments part of ship.

'B' means the breadth of the ship, which is the last value of '10 station' for most of ships.



```python
def Hydrostatic_curve():
    w = dataoffset.shape[1]-3
    h = WL(7, -1) - WL(6, -1)
    f = 1
    t = -WL(-1, -1)
    j = WL(25, -1) - WL(24, -1)
    L = WL(24, -1)
    B = WL(12, w)
    data = WLlist(w, h, f, L, B)
    y = range(2, w)
    plt.yticks(range(2, w, 2))
    plt.xticks(range(-5, 150, 5))
    x = [DIS(a + 1, h, f, t, j) / 8000 for a in y]
    z = [DIS(a + 1, h, f, t, j) / 8000 / 1.025 for a in y]
    e = [KB(a + 1, h, f, t, j) for a in y]
    g = [Aw(a + 1, h) / 600 for a in y]
    p = [TPC(a + 1, h) / 20 for a in y]
    u = [Cm(a + 1, B) / 0.05 for a in y]
    c = [Cw(a + 1, h, L, B) / 0.05 for a in y]
    q = [KM(a + 1, h, f, t, j) / 2 for a in y]
    m = [MTC(a + 1, h, f, t, j, L) / 200 for a in y]
    v = [KML(a + 1, h, f, t, j, L) / 50 for a in y]
    ww = [Cb(a + 1, h, f, L, B, t, j) / 0.05 for a in y]
    ii = [Cp(a + 1, h, f, L, B, t, j) / 0.05 for a in y]
    kk = [SSLCF(a + 1, h, t, j, L) for a in y]
    hh = [LCB(a + 1, h, f, t, j, L) for a in y]
    ll = [Cvp(a + 1, h, f, L, B, t, j) / 0.05 for a in y]
```



```
    plt.plot(x, y, 'r')
    plt.plot(z, y, 'b')
    plt.plot(e, y, 'g')
    plt.plot(g, y, 'c')
    plt.plot(p, y, 'm')
    plt.plot(u, y, 'k')
    plt.plot(c, y, 'y')
    plt.plot(q, y, color='brown')
    plt.plot(m, y, color='grey')
    plt.plot(v, y, color='violet')
    plt.plot(ww, y, color='orange')
    plt.plot(ii, y, '--')
    plt.plot(kk, y, color='tan')
    plt.plot(hh, y, color='maroon')
    plt.plot(ll, y, color='navy')
    plt.grid()
    plt.legend(
        ['△ 1= 8000t', '▽ 1= 8000t', 'KB 1= 1m', 'Aw 1= 600m^2', 'TPC 1= 20t', 'Cm 1= 0.05', 'Cw 1= 0.05', 'KM 1= 2m',
         'MTC 1= 200t-m', 'KML 1= 50m', 'Cb 1= 0.05', 'Cp 1= 0.05', 'LCF 1= 1m', 'LCB 1= 1m', 'Cvp 1= 0.05'])

    return plt.show()
```

The command lines above show the ‘hyprostatic curve’ by using the 'matplotlib.pyplot' library. Library 'matplotlib.pyplot' has been entered as 'plt' for use. The functions in this library, ‘.yticks’ and ‘.xtkiks’, are required in the form of each (start point, end point, interval) of the graph, so that you can set each value. 'range' is an arrangement from 2 to the last number of waterline, which is the y-axis. The x-axis for each value is created by 'Variables = [Functions]'. We used command function 'for' in '[]' to add the value up to the last waterline as accumulated starting with waterline no.2. Also, use the '.plot' function from the library (x-axis value, y-axis value, color or shape of the line). Multiple curves are visible in one window, with different colors for each curve. We also used the 'grid()' function to put a grid so that users can easily read values, and also the 'legend()' function to explain what each graph means. Finally, we can use the command function ‘show()’ to see curve by users. Main dimensions are automatically imported from the offset table, just like ship calculation table.

3.2 GUI form

Another method to get the calculation table and hydrostatic curves of ship is using GUI form of ARPA-GO. This form is convenient to implement and can receive clear results.

3.2.1 Main Window

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is Python itself library. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

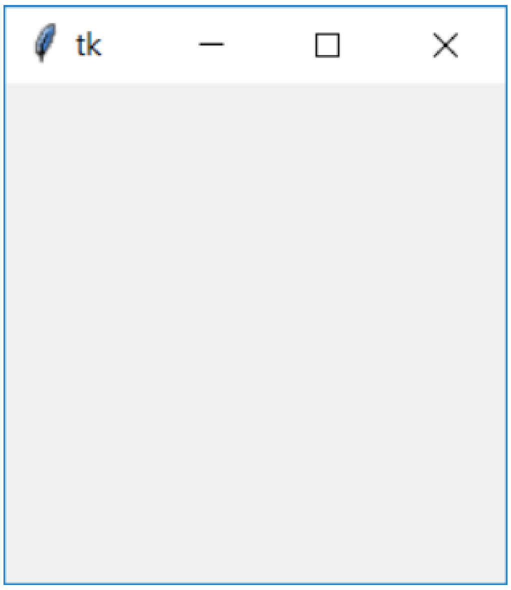
```
import tkinter
```

Python include the GUI module using the top command. You can use the tkinter function by using 'tkinter.*'.

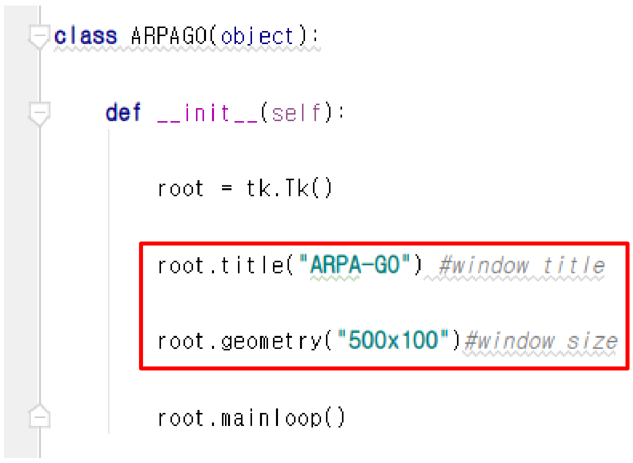
```
import tkinter

window=tkinter.Tk()

window.mainloop()
```

In this command, 'window' refers to the name of the execution window and can be declared by any name users want. The red command allows you to create the highest level of windowing. In some cases, users can writer tk (or ttk) which is shortcut of tkinter for example window = tk.Tk( ).

The last command means to run until the window ends. Similarly, you can apply any name of the execution window instead of window.

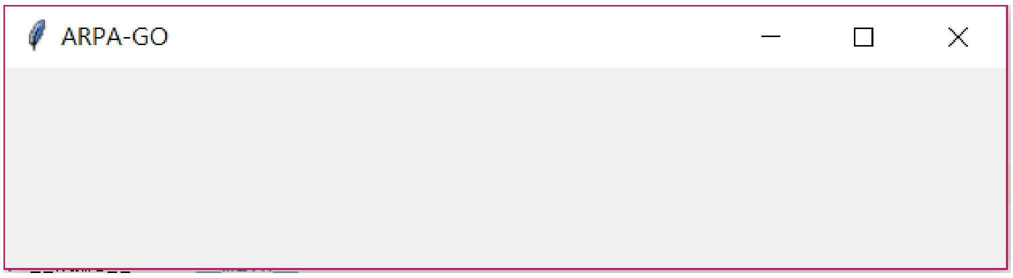


When we execute this, the most basic window is created.



```
class ARPAGO(object):
    def __init__(self):
        root = tk.Tk()
        root.title("ARPA-GO") #window title
        root.geometry("500x100")#window size
        root.mainloop()
```

This is the coding of our project window. We've named window as "ARPA-GO" which is our project name, and set the proper size.



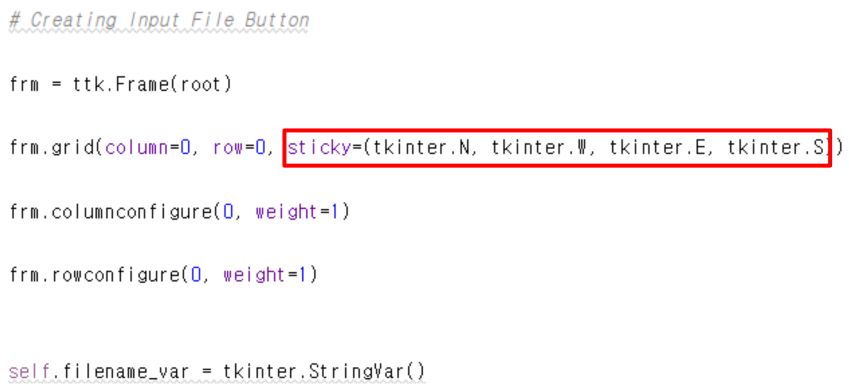
When we run this coding, we are available to get the main window of our project.

3.2.2 Import Excel

As with coding, it require to import the Excel file which contains offset table of the ship users want to know. In GUI format, you import offset table by using open file path.

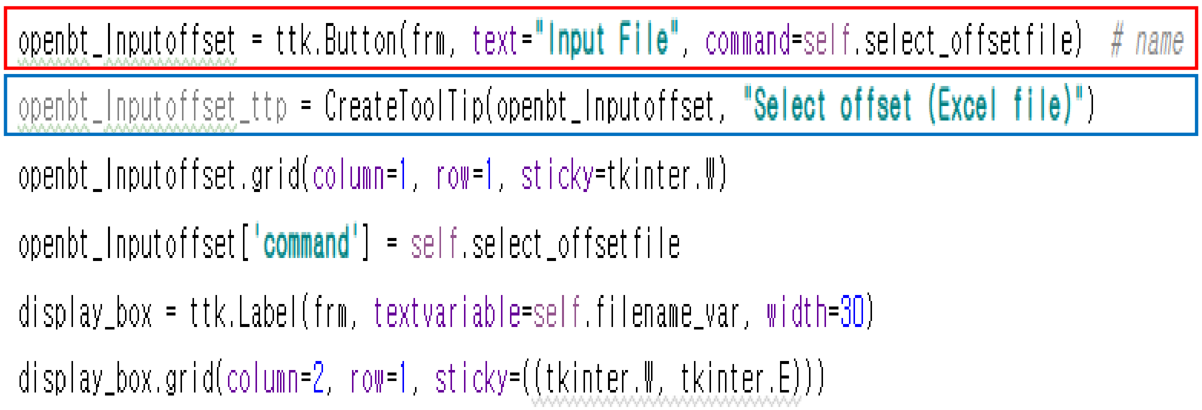
To do that, we have to use button widget. The Button widget is a standard Tkinter widget, which is used for various kinds of buttons. A button is a widget which is designed for the user to interact with, i.e. if the button is pressed by mouse click some action might be started. They can also contain text and images like labels. While labels can display text in various fonts, a button can only display text in a single font. The text of a button can span more than one line.

A Python function or method can be associated with a button. This function or method will be executed, if the button is pressed in some way. Our project ARPA-GO used this function of button widget to get the results.



```
# Creating Input File Button

frm = ttk.Frame(root)

frm.grid(column=0, row=0, sticky=(tkinter.N, tkinter.W, tkinter.E, tkinter.S))

frm.columnconfigure(0, weight=1)

frm.rowconfigure(0, weight=1)


self.filename_var = tkinter.StringVar()
```

To create a button, users have to declare a function for the location of the executable window. The part of the red box is the coding that declared the function by the location of the north, west, east and south side of the window that was created earlier.



```
openbt_Inputoffset = ttk.Button(frm, text="Input File", command=self.select_offsetfile)  # name
openbt_Inputoffset_ttp = CreateToolTip(openbt_Inputoffset, "Select offset (Excel file)")
openbt_Inputoffset.grid(column=1, row=1, sticky=tkinter.W)
openbt_Inputoffset['command'] = self.select_offsetfile
display_box = ttk.Label(frm, textvariable=self.filename_var, width=30)
display_box.grid(column=2, row=1, sticky=((tkinter.W, tkinter.E)))
```

Similar to creating a main window, create buttons using widgets and set the name, size and location of the buttons. The command function corresponding to this button is named open_button, and the text to be written in the button is "Input File".

In the red box, there is command lines like command = self.selcet_file. This means that when you click this button, you follow the class named self_file. We will discuss Python's Class features in more detail in the appendix

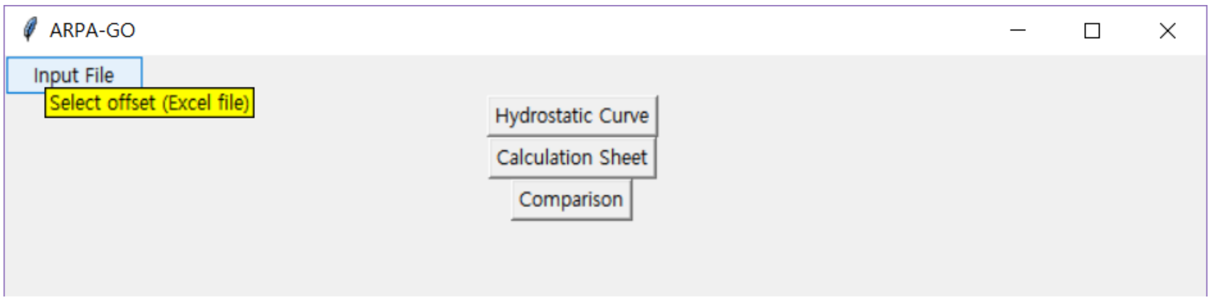
To use button's function more effectively, we used tooltip. A tooltip is often used to specify extra information about something when the user moves the mouse pointer over an element. Using this effect of tooltip, we explained button briefly. Corresponding command lines are the coding above which is in blue box.

After making button successfully, we have to add function which will be conducted along with button click.

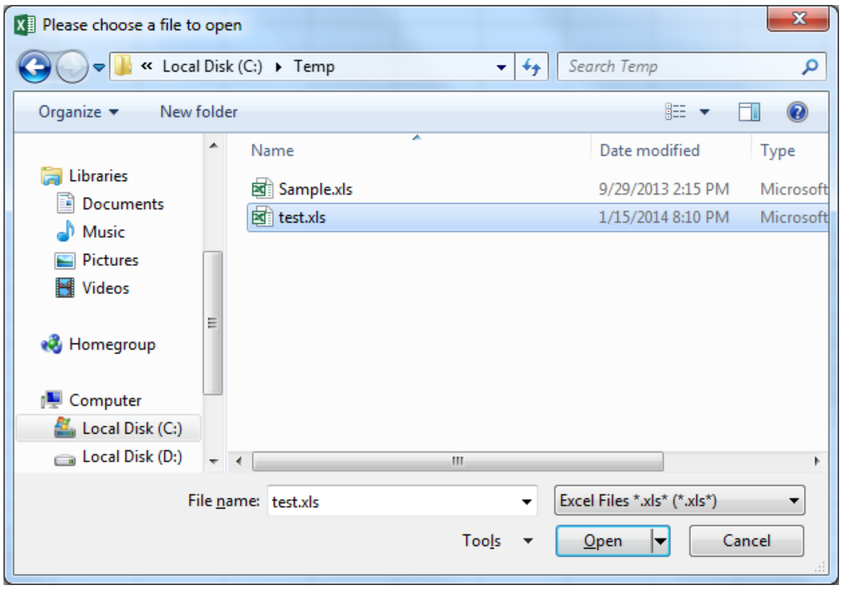
```
# Import offset file

def select_offsetfile(self):
    global dataoffset
    filename = filedialog.askopenfilename(
        filetypes=[('Excel Spreadsheet', '*.xlsx'), ('Excel Spreadsheet', '*.xls'), ('All files', '*.*')])
    self.filename_var.set(filename)
    dataoffset = pd.read_excel(filename)
```

The function that corresponds to this button is importing the file. 'filename' is a command declared to import an offset file. You can also select extensions depending on the type of file. As our project proceeds with the calculation through the Excel file which contains the offset of the ship, the extension of Excel file has been prepared.

The results for the above coding are as follows:



The tooltip provides a brief description of the Input File button.



Name	Date modified	Type
Sample.xls	9/29/2013 2:15 PM	Microsoft
test.xls	1/15/2014 8:10 PM	Microsoft

File name: test.xls

Excel Files *.xls* (*.xls*)

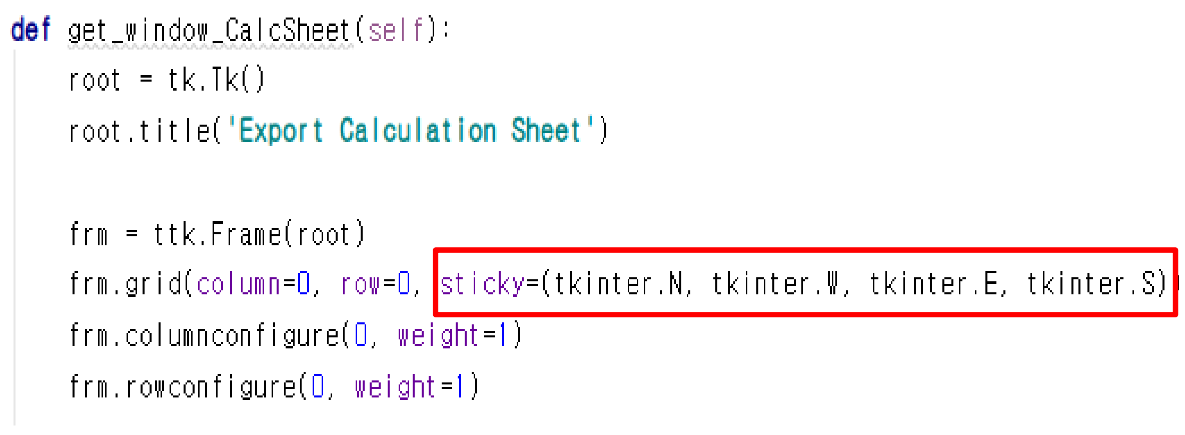
By clicking the button, you are able to import the offset file stored on your device.

3.2.3 Ship calculation sheet

It is also same as Input File. We create button first and put the function to work along with certain button

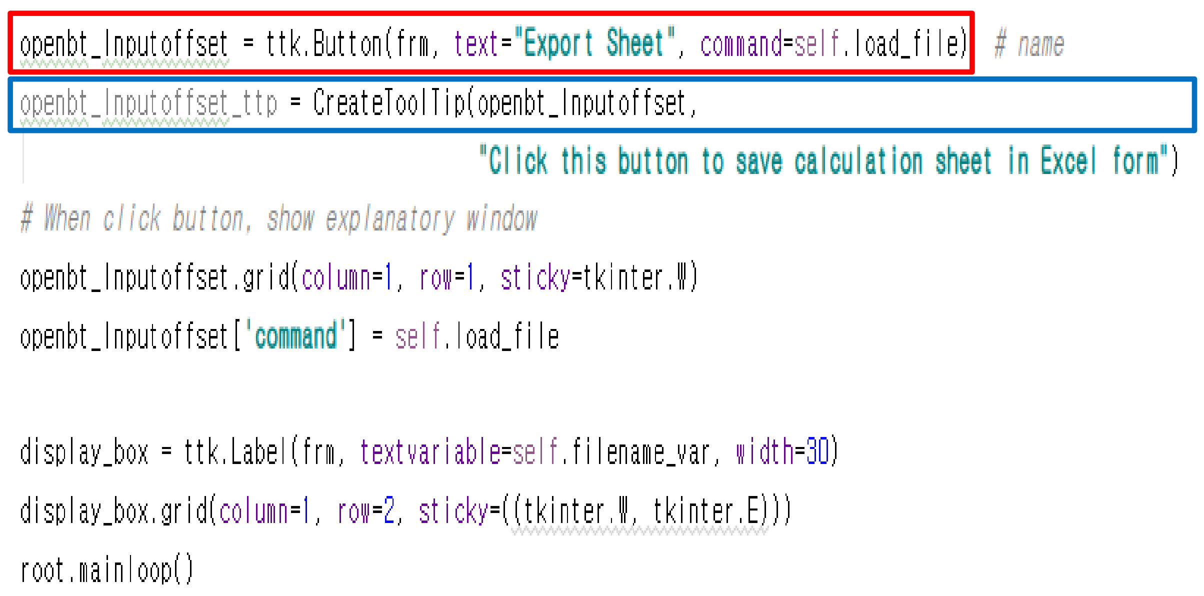
```
# Calculation Sheet Button
bt_CalcSheet = tk.Button(root, text="Calculation Sheet", command=self.get_window_CalcSheet)
bt_CalcSheet.grid(column=1, row=6)
```

Create button called 'Calculation Sheet' and set the positon of the button.



```python
def get_window_CalcSheet(self):
    root = tk.Tk()
    root.title('Export Calculation Sheet')

    frm = ttk.Frame(root)
    frm.grid(column=0, row=0, sticky=(tkinter.N, tkinter.W, tkinter.E, tkinter.S))
    frm.columnconfigure(0, weight=1)
    frm.rowconfigure(0, weight=1)
```

Just same as before, we declared the function by the location of the north, west, east and south side of the window.



```python
openbt_Inputoffset = ttk.Button(frm, text="Export Sheet", command=self.load_file)  # name
openbt_Inputoffset_ttp = CreateToolTip(openbt_Inputoffset,
                                       "Click this button to save calculation sheet in Excel form")
# When click button, show explanatory window
openbt_Inputoffset.grid(column=1, row=1, sticky=tkinter.W)
openbt_Inputoffset['command'] = self.load_file

display_box = ttk.Label(frm, textvariable=self.filename_var, width=30)
display_box.grid(column=1, row=2, sticky=((tkinter.W, tkinter.E)))
root.mainloop()
```

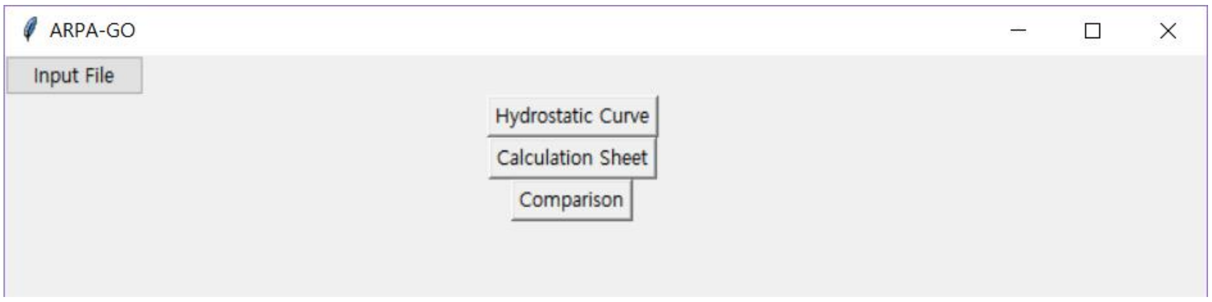
Then we have to declare the button name and class that have to follow. After that, the tooltip for this button and the location should be decided.

```
# Export Calculation Sheet

def export_calcsheet(self):
    file = Calculation_sheet()

    savefile = filedialog.asksaveasfilename(filetypes=(("Excel files", "*.xlsx"), ("All files", "*.*")))
    file.to_excel(savefile + ".xlsx", index=False, sheet_name="Results")
```

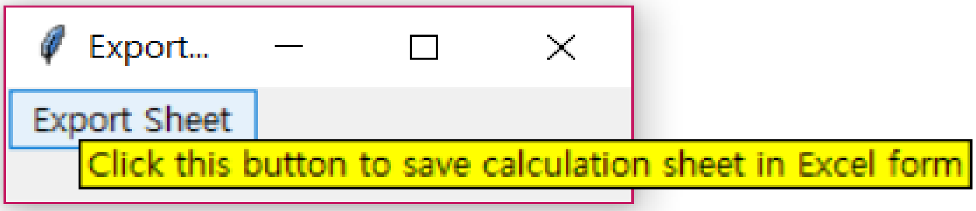
It's turn to add function. You can save a ship calculation table as an Excel file which is calculated from the data obtained from the offset. It allows you to create an Excel file for a quick, at-a-glance view of the calculations. By changing the extension of the file, it is possible to save it in other file formats.

After adding the function, the coding for this part is completed.

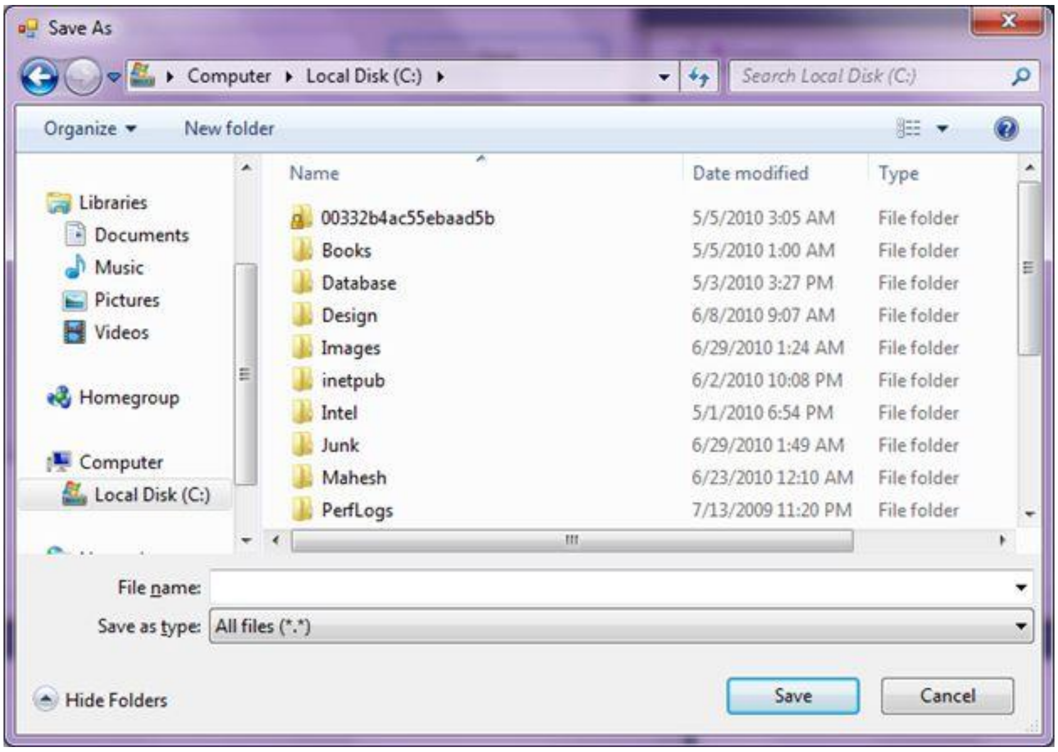
The results for the above coding are as follows:



Button called 'Calculation Sheet' is created.



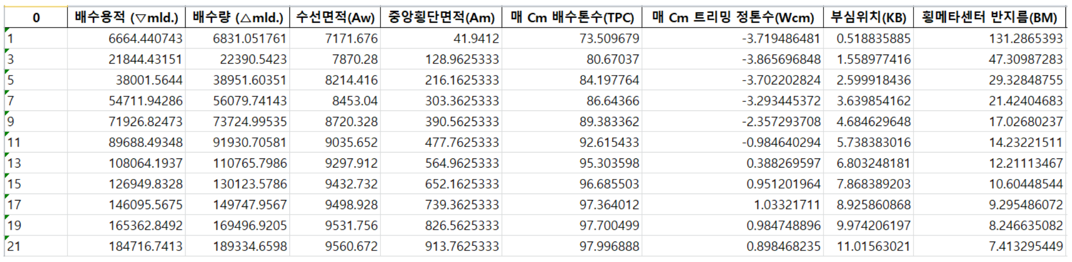
The tooltip provides a brief description of the Export Sheet button.



Name	Date modified	Type
00332b4ac55ebaad5b	5/5/2010 3:05 AM	File folder
Books	5/5/2010 1:00 AM	File folder
Database	5/3/2010 3:27 PM	File folder
Design	6/8/2010 9:07 AM	File folder
Images	6/29/2010 1:24 AM	File folder
inetpub	6/2/2010 10:08 PM	File folder
Intel	5/1/2010 6:54 PM	File folder
Junk	6/29/2010 1:49 AM	File folder
Mahesh	6/23/2010 12:10 AM	File folder
PerfLogs	7/13/2009 11:20 PM	File folder

File name:
Save as type: All files (*.*)

By clicking the button, you are able to save the calculation table of ship on your device.



0	배수용적 (▽mld.)	배수량 (△mld.)	수선면적(Aw)	중앙횡단면적(Am)	매 Cm 배수톤수(TPC)	매 Cm 트리밍 정톤수(Wcm)	부심위치(KB)	횡메타센터 반지름(BM)
1	6664.440743	6831.051761	7171.676	41.9412	73.509679	-3.719486481	0.518835885	131.2865393
3	21844.43151	22390.5423	7870.28	128.9625333	80.67037	-3.865696848	1.558977416	47.30987283
5	38001.5644	38951.60351	8214.416	216.1625333	84.197764	-3.702202824	2.599918436	29.32848755
7	54711.94286	56079.74143	8453.04	303.3625333	86.64366	-3.293445372	3.639854162	21.42404683
9	71926.82473	73724.99535	8720.328	390.5625333	89.383362	-2.357293708	4.684629648	17.02680237
11	89688.49348	91930.70581	9035.652	477.7625333	92.615433	-0.984640294	5.738383016	14.23221511
13	108064.1937	110765.7986	9297.912	564.9625333	95.303598	0.388269597	6.803248181	12.21113467
15	126949.8328	130123.5786	9432.732	652.1625333	96.685503	0.951201964	7.868389203	10.60448544
17	146095.5675	149747.9567	9498.928	739.3625333	97.364012	1.03321711	8.925860868	9.295486072
19	165362.8492	169496.9205	9531.756	826.5625333	97.700499	0.984748896	9.974206197	8.246635082
21	184716.7413	189334.6598	9560.672	913.7625333	97.996888	0.898468235	11.01563021	7.413295449

When you click the 'Calculation Sheet' button in the execution window, the calculation table of the ship is saved to the device as an Excel file. This allows you to check the ship's calculation table in detail and can see at a glance the result of calculation in certain station and waterline without hydrostatic curves. The value shown above is only a small part of ARPA-GO. Our project can check 20 values of ship calculation.

3.2.4 Hydrostatic curves

Drawing Hydrostatic curves is relatively simple.

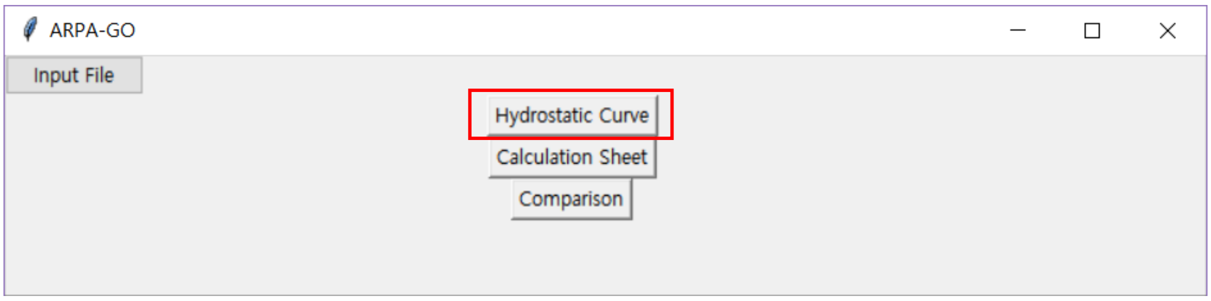
```
# Hydrostatic Curve Button
bt_HydroC = tk.Button(root, text="Hydrostatic Curve", command=self.get_window_hydrostaticC)
bt_HydroC.grid(column=1, row=5)  # Location
```

We specified the name, class, and location of the button in the same way as the previous buttons.

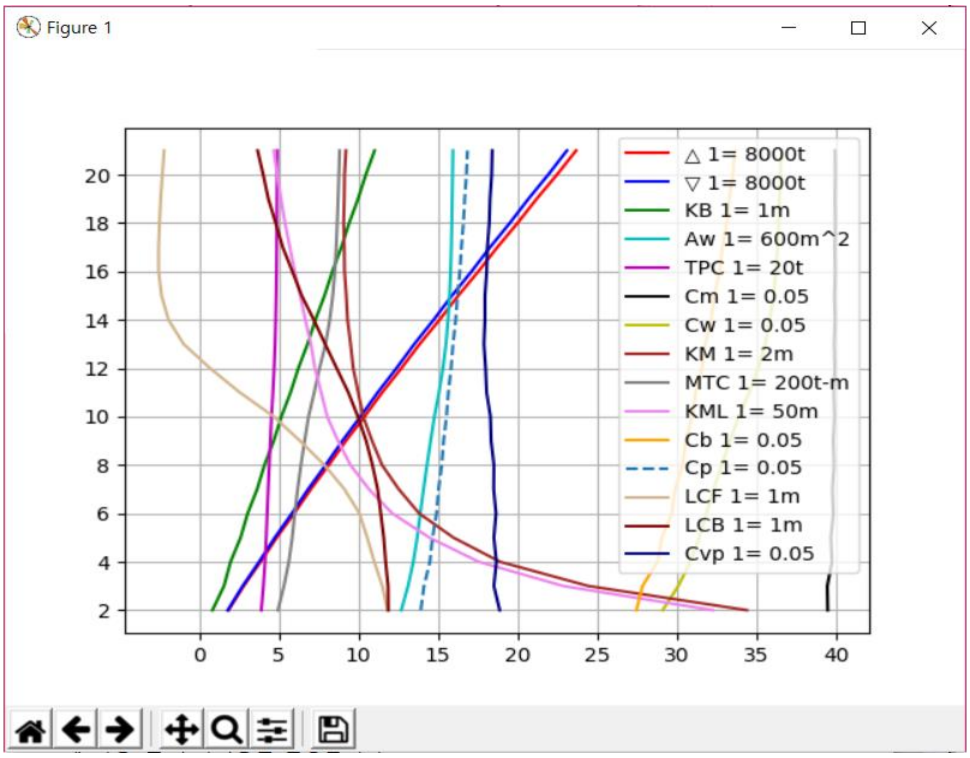
```
# 'Hydrostatic Curve' button click event

def get_window_hydrostaticC(self):
    fig = Figure()
    Hydrostatic_curve()
```

The command function that creates a window which represents a graph is Figure( ). We write a graph which will appear in this window. In this window, the hydrostatic curve appears and can be saved.

The results for the above coding are as follows:



Click the button 'Hydrostatic Curve'.



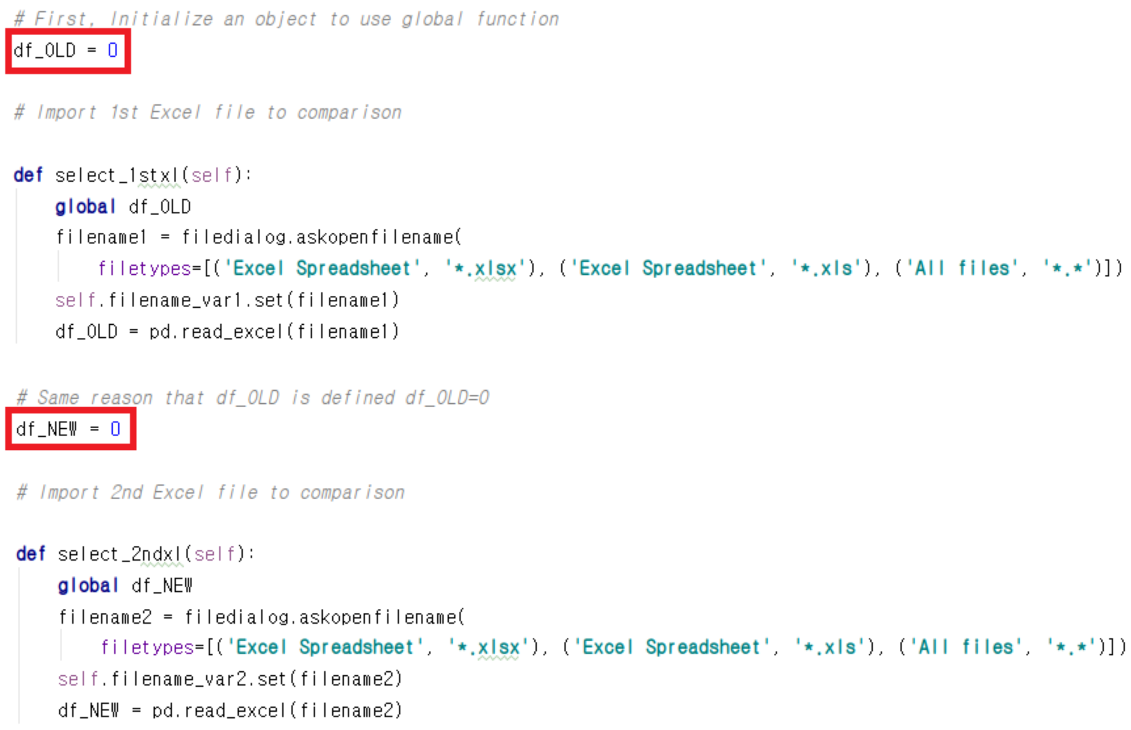
As a result, the hydrostatic curve of the ship is shown.

3.2.5 Comparison of Results

First of all, This figure is a command to create a new Excel showing different values between two excel files.

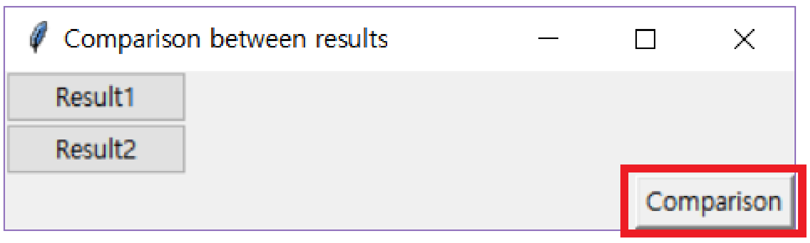
```
def excel_diff(path_OLD, path_NEW, index_col):
    df_OLD = pd.read_excel(path_OLD, index_col=index_col).fillna(0)
    df_NEW = pd.read_excel(path_NEW, index_col=index_col).fillna(0)

    # 차이점 나타내기
    dfDiff = df_NEW.copy()
    newRows = []

    cols_OLD = df_OLD.columns
    cols_NEW = df_NEW.columns
    sharedCols = list(set(cols_OLD).intersection(cols_NEW))

    for row in dfDiff.index:
        if (row in df_OLD.index) and (row in df_NEW.index):
            for col in sharedCols:
                value_OLD = df_OLD.loc[row, col]
                value_NEW = df_NEW.loc[row, col]
                if value_OLD == value_NEW:
                    dfDiff.loc[row, col] = df_NEW.loc[row, col]
                else:
                    dfDiff.loc[row, col] = ('{}→{}').format(value_OLD, value_NEW)
        else:
            newRows.append(row)
```

df_OLD and df_NEW are the Excel files's path received from Result1 and Result2.

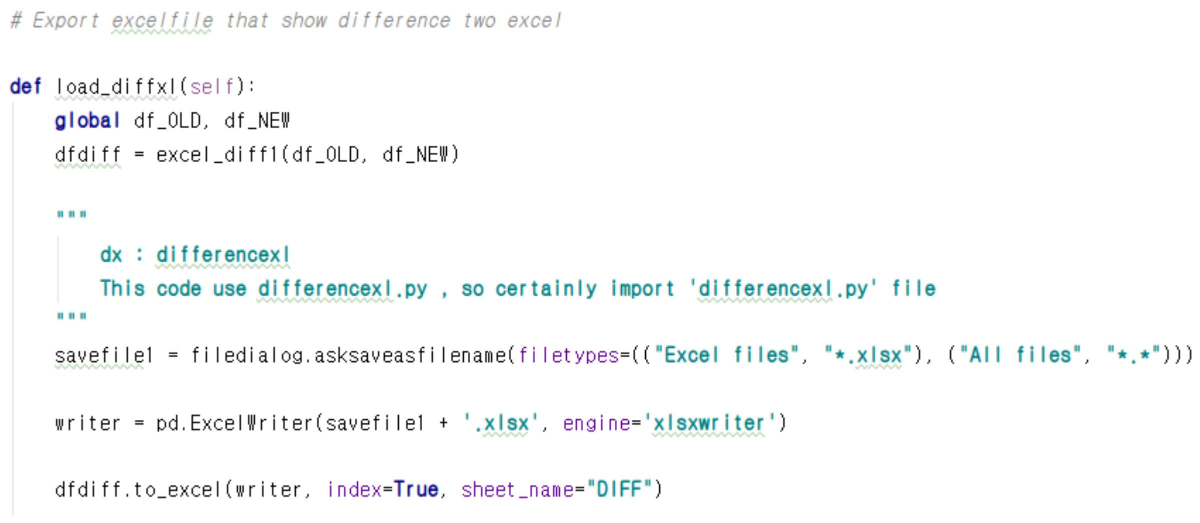


```
# First, Initialize an object to use global function
df_OLD = 0

# Import 1st Excel file to comparison

def select_1stxl(self):
    global df_OLD
    filename1 = filedialog.askopenfilename(
        filetypes=[('Excel Spreadsheet', '*.xlsx'), ('Excel Spreadsheet', '*.xls'), ('All files', '*.*')])
    self.filename_var1.set(filename1)
    df_OLD = pd.read_excel(filename1)

# Same reason that df_OLD is defined df_OLD=0
df_NEW = 0

# Import 2nd Excel file to comparison

def select_2ndxl(self):
    global df_NEW
    filename2 = filedialog.askopenfilename(
        filetypes=[('Excel Spreadsheet', '*.xlsx'), ('Excel Spreadsheet', '*.xls'), ('All files', '*.*')])
    self.filename_var2.set(filename2)
    df_NEW = pd.read_excel(filename2)
```

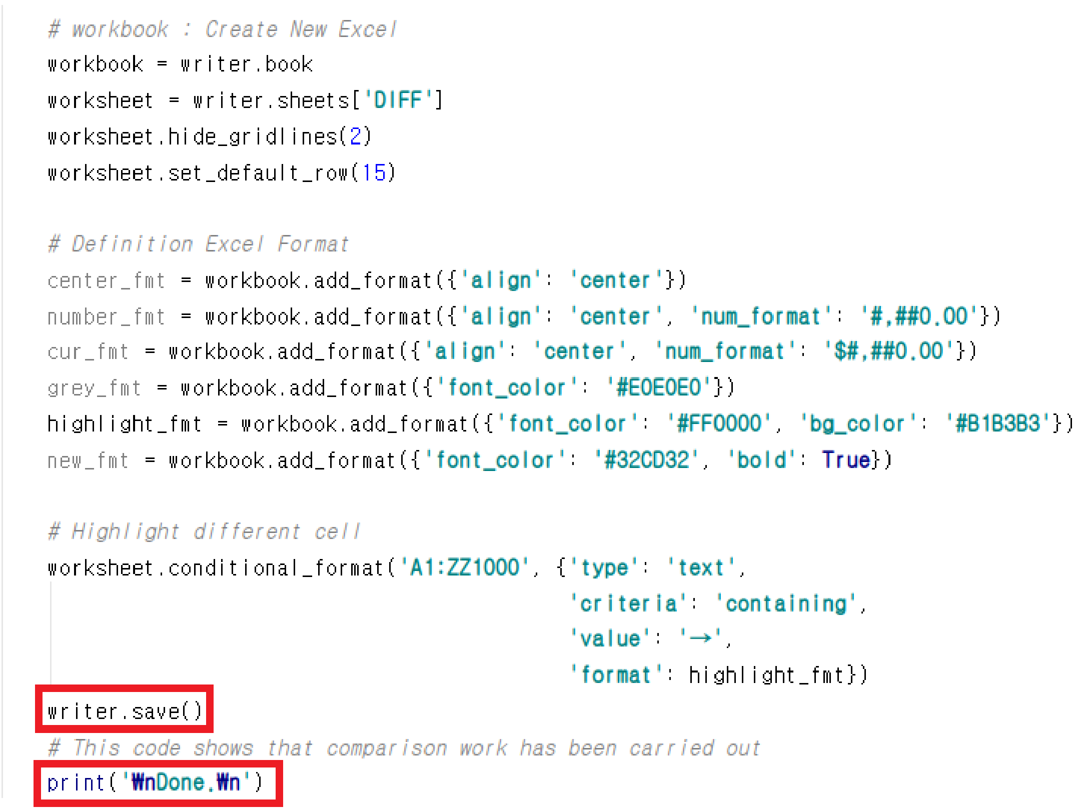
In the command lines, df_OLD=0, df_NEW=0 are written. This is because they must be declared in order to use it as a global function.



Use 'Result1' and 'Result2' to receive an Excel file and then click the 'Comparison' button to execute the following code.



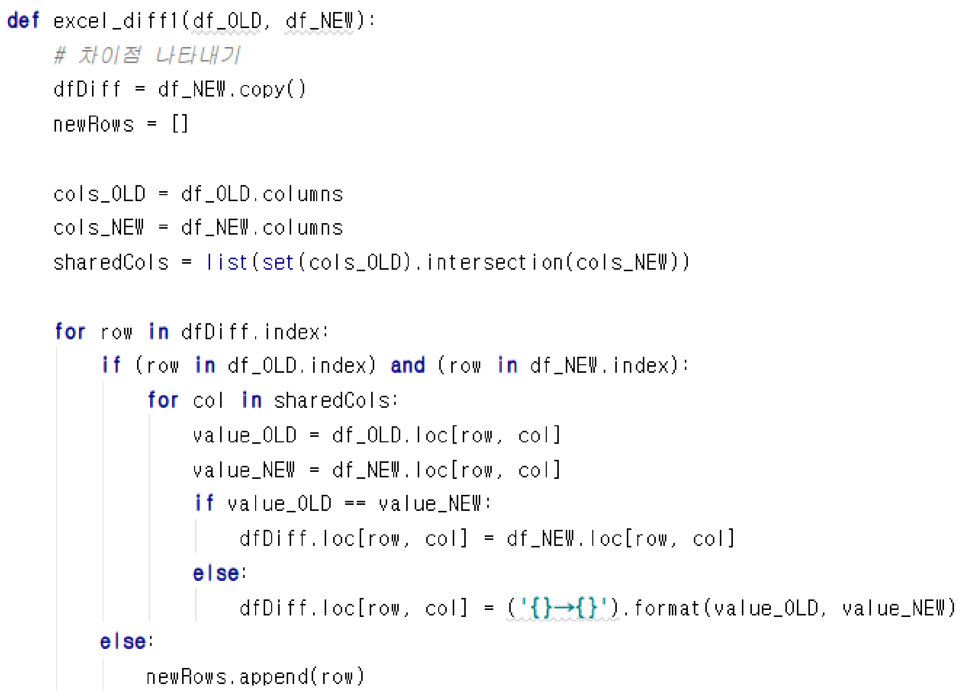
```
# Export excelfile that show difference two excel

def load_diffxl(self):
    global df_OLD, df_NEW
    dfdiff = excel_diff1(df_OLD, df_NEW)

    """
        dx : differencexl
        This code use differencexl.py , so certainly import 'differencexl.py' file
    """
    savefile1 = filedialog.asksaveasfilename(filetypes=(("Excel files", "*.xlsx"), ("All files", "*.*")))

    writer = pd.ExcelWriter(savefile1 + '.xlsx', engine='xlsxwriter')

    dfdiff.to_excel(writer, index=True, sheet_name="DIFF")
```



```python
# workbook : Create New Excel
workbook = writer.book
worksheet = writer.sheets['DIFF']
worksheet.hide_gridlines(2)
worksheet.set_default_row(15)

# Definition Excel Format
center_fmt = workbook.add_format({'align': 'center'})
number_fmt = workbook.add_format({'align': 'center', 'num_format': '#,##0.00'})
cur_fmt = workbook.add_format({'align': 'center', 'num_format': '$#,##0.00'})
grey_fmt = workbook.add_format({'font_color': '#E0E0E0'})
highlight_fmt = workbook.add_format({'font_color': '#FF0000', 'bg_color': '#B1B3B3'})
new_fmt = workbook.add_format({'font_color': '#32CD32', 'bold': True})

# Highlight different cell
worksheet.conditional_format('A1:ZZ1000', {'type': 'text',
                                            'criteria': 'containing',
                                            'value': '→',
                                            'format': highlight_fmt})
writer.save()
# This code shows that comparison work has been carried out
print('₩nDone.₩n')
```

Writer.save() : Save the Excel format specified by the Definition Excel Format and Highlight differential cell.

Print('₩nDone.₩n') : The user will know that an Excel file has been created.



```python
def excel_diff1(df_OLD, df_NEW):
    # 차이점 나타내기
    dfDiff = df_NEW.copy()
    newRows = []

    cols_OLD = df_OLD.columns
    cols_NEW = df_NEW.columns
    sharedCols = list(set(cols_OLD).intersection(cols_NEW))

    for row in dfDiff.index:
        if (row in df_OLD.index) and (row in df_NEW.index):
            for col in sharedCols:
                value_OLD = df_OLD.loc[row, col]
                value_NEW = df_NEW.loc[row, col]
                if value_OLD == value_NEW:
                    dfDiff.loc[row, col] = df_NEW.loc[row, col]
                else:
                    dfDiff.loc[row, col] = ('{}→{}').format(value_OLD, value_NEW)
        else:
            newRows.append(row)
```

Excel_diff1 cannot recognize by load_diffxl because Excel_diff uses three variables.

To resolve this, We've create a function Excel_diff1 function of two variables.

4. Appendix

This chapter describes the additional functions used in ARPA-GO.

4.1 Additional Function

4.1.1 Lisp

VisualLisp (AutoLisp) is a program language supported by AutoCAD that enables users to develop and use the necessary functions on their own. This is derived from the program language 'LISP' and it is a representative case of practical use of LISP. VisualLisp files can be created or modified using the 'Notepad' or 'Manage-Applications-Visual LISP Editor' in CAD. If you write the code in Notepad, you have to save file extension as '.LSP'

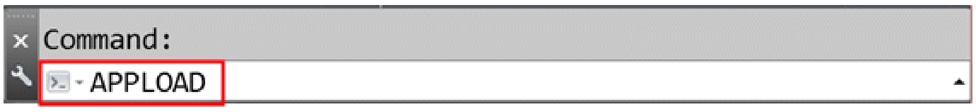
When the user performs ship calculation using ARPA-GO, we provide coordinate extraction function that allows users to work more efficiently. ARPA-GO Lisp, a coordinate extraction function, is created using the Visual LISP Editor and it can be modified at your convenience.

"ARPA-GO Lisp" consists of a main function that actually executes the lisp and several sub-functions that assist main function. (Sub-function can consist of other sub-functions.) If you want to modify Visual Lisp to suit your convenience, you need to know the meaning of each function phrase. However, there is not enough data in Korean to study Visual Lisp. Therefore, we have attached a detailed description of each function of ARPA-GO Lisp in the hope that it will help to activate Visual Lisp in Korean.

How to use "ARPA-GO Lisp"

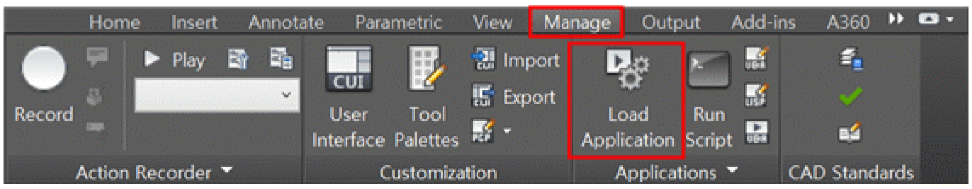
1) Activate the Body Plan of ship you want to calculate.

2) Type a command 'Appload' or select 'Manage-Applications-Load Application'.



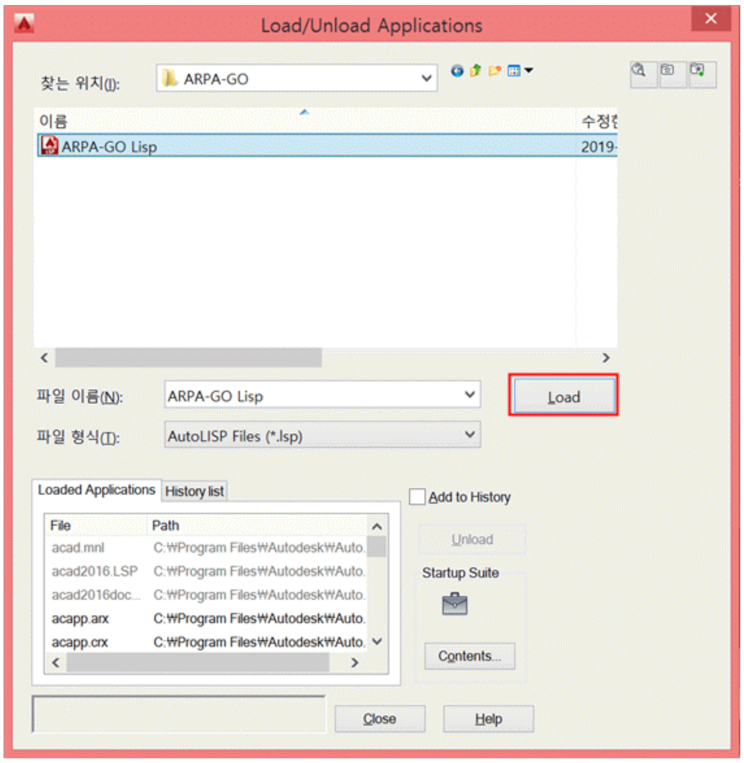
```
Command:
APPLOAD
```

[Type a command]



[Load Application]

3) When the ‘Load/Unload Applications’ window pops up, click the ‘Load’ button.



찾는 위치(I): ARPA-GO

이름	수정
ARPA-GO Lisp	2019-

파일 이름(N): ARPA-GO Lisp

파일 형식(T): AutoLISP Files (*.lsp)

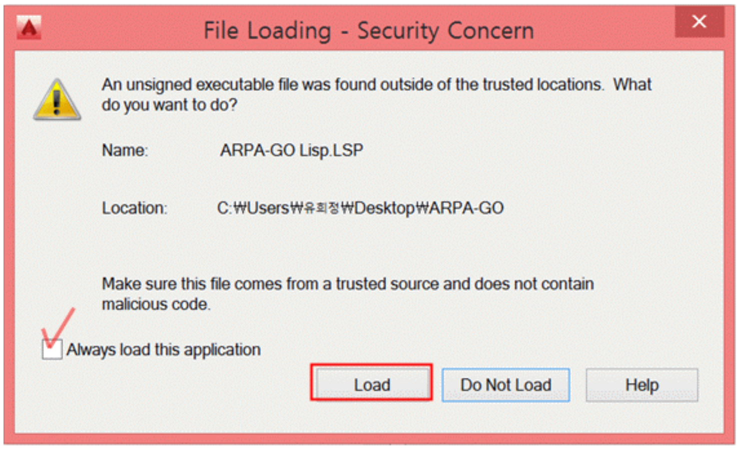
Loaded Applications | History list

Add to History

File	Path
acad.mnl	C:₩Program Files₩Autodesk₩Auto.
acad2016.LSP	C:₩Program Files₩Autodesk₩Auto.
acad2016doc...	C:₩Program Files₩Autodesk₩Auto.
acapp.arx	C:₩Program Files₩Autodesk₩Auto.
acapp.crx	C:₩Program Files₩Autodesk₩Auto.

4) If ‘security concern’ window pops up, click the ‘Load’ button.

(It doesn’t matter to check the box ‘Always load this application’.)



An unsigned executable file was found outside of the trusted locations. What do you want to do?

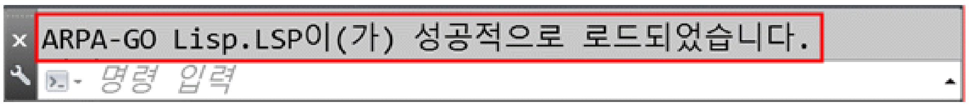
Name: ARPA-GO Lisp.LSP

Location: C:₩Users₩유희정₩Desktop₩ARPA-GO

Make sure this file comes from a trusted source and does not contain malicious code.

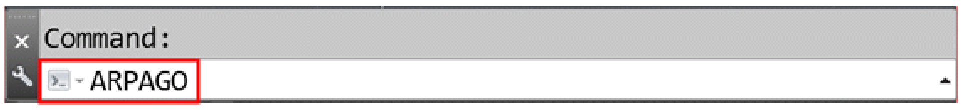
[✓] Always load this application

5) After click the close button, you are confirmed that the ARPA-GO Lisp has been successful ly loaded through the command line.



```
ARPA-GO Lisp.LSP이(가) 성공적으로 로드되었습니다.
명령 입력
```

6) Type a command 'ARPAGO' or 'arpago'.

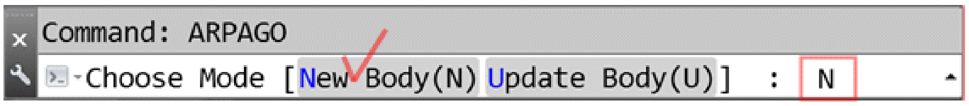


```
Command:
ARPAGO
```

7) There are two modes of coordinate extraction, a 'New Body (N)' mode to extract coordinate s of new hull form and 'Update Body (U)' mode to extract coordinates of modified hull fo rm.

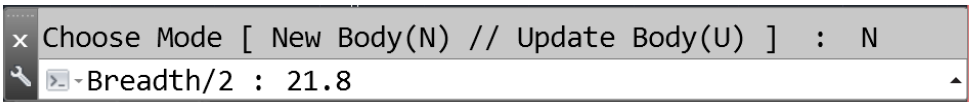
- *First, let us explain the 'New Body(N)' mode.*

01. Click 'New Body(N)' on the command line or enter 'N'.



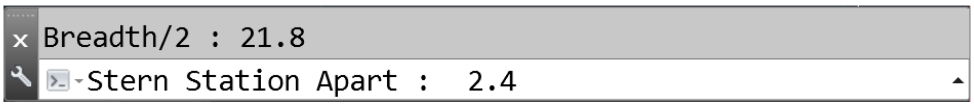
```
Command: ARPAGO
Choose Mode [New Body(N) Update Body(U)]  : N
```

02. Enter the values accordingly as shown in the command line.



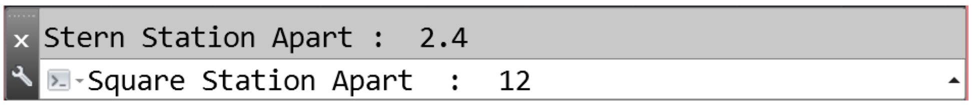
```
Choose Mode [ New Body(N) // Update Body(U) ]  :  N
Breadth/2 : 21.8
```

[Half Breadth]



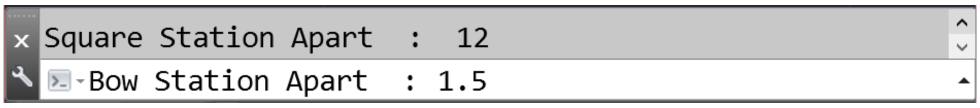
```
Breadth/2 : 21.8
Stern Station Apart :  2.4
```

[Stern Station Spacing]



```
Stern Station Apart :  2.4
Square Station Apart  :  12
```

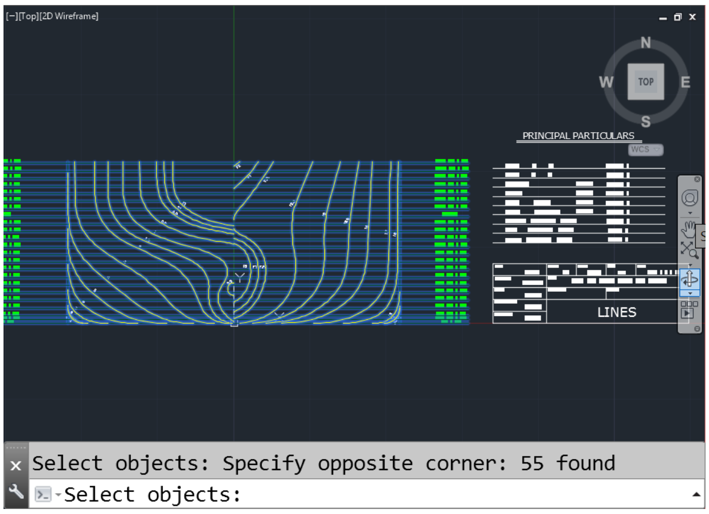
[Square Station Spacing]



```
Square Station Apart  :  12
Bow Station Apart  : 1.5
```

[Bow Station Spacing]

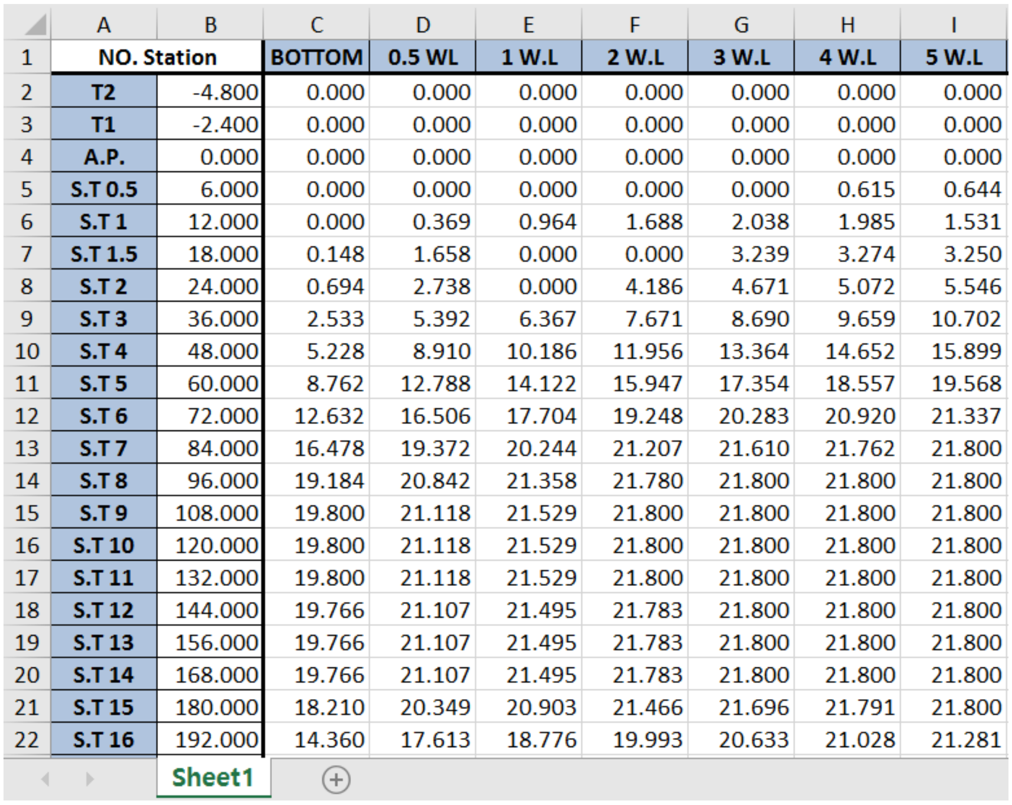
03. Drag the entire Body Plan (hull form, waterline, etc.). It doesn't matter if it contains te xt, vertical line, etc.



```
Select objects: Specify opposite corner: 55 found
Select objects:
```

04. When pressing Enter after all objects are selected, alert window comes out to indicate that it is complete. And you can check the extracted coordinate values via Excel.

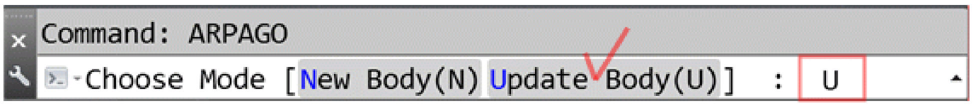
05. The offset table completed through ARPA-GO Lisp is created as shown in the figure b elow.



	A	B	C	D	E	F	G	H	I
1	NO. Station		BOTTOM	0.5 WL	1 W.L	2 W.L	3 W.L	4 W.L	5 W.L
2	T2	-4.800	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	T1	-2.400	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	A.P.	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	S.T 0.5	6.000	0.000	0.000	0.000	0.000	0.000	0.615	0.644
6	S.T 1	12.000	0.000	0.369	0.964	1.688	2.038	1.985	1.531
7	S.T 1.5	18.000	0.148	1.658	0.000	0.000	3.239	3.274	3.250
8	S.T 2	24.000	0.694	2.738	0.000	4.186	4.671	5.072	5.546
9	S.T 3	36.000	2.533	5.392	6.367	7.671	8.690	9.659	10.702
10	S.T 4	48.000	5.228	8.910	10.186	11.956	13.364	14.652	15.899
11	S.T 5	60.000	8.762	12.788	14.122	15.947	17.354	18.557	19.568
12	S.T 6	72.000	12.632	16.506	17.704	19.248	20.283	20.920	21.337
13	S.T 7	84.000	16.478	19.372	20.244	21.207	21.610	21.762	21.800
14	S.T 8	96.000	19.184	20.842	21.358	21.780	21.800	21.800	21.800
15	S.T 9	108.000	19.800	21.118	21.529	21.800	21.800	21.800	21.800
16	S.T 10	120.000	19.800	21.118	21.529	21.800	21.800	21.800	21.800
17	S.T 11	132.000	19.800	21.118	21.529	21.800	21.800	21.800	21.800
18	S.T 12	144.000	19.766	21.107	21.495	21.783	21.800	21.800	21.800
19	S.T 13	156.000	19.766	21.107	21.495	21.783	21.800	21.800	21.800
20	S.T 14	168.000	19.766	21.107	21.495	21.783	21.800	21.800	21.800
21	S.T 15	180.000	18.210	20.349	20.903	21.466	21.696	21.791	21.800
22	S.T 16	192.000	14.360	17.613	18.776	19.993	20.633	21.028	21.281

- *Next, let us explain the 'Update Body(U)' mode.*

01. Click 'Update Body(U)' on the command line or type 'U'.



```
Command: ARPAGO
Choose Mode [New Body(N) Update Body(U)] : U
```

02. A complication alert window will appear immediately without having to select an object. And a new Excel file is created for the modified hull form.

8) By saving the Excel file's name and the path you want, ARPA-GO Lisp completes the extraction of coordinates.

"ARPA-GO Lisp": functions

Before we explain the syntax, we would like to describe the AutoListp itself function used. Because AutoLisp is expressed in parentheses, an error occurs if the number of left and right parentheses is different. Also, there must be only one function in one parenthesis, and the expression is not case-sensitive in English.

Type	***Color***
Built-In Function and Symbol	**Blue**
String	**Pink**
Integer	**Green**
Real number	**Cyan**
Annotation	**Gray-Based Pink**
Parentheses	**Red**
Variable / Unrecognizable element	**Black**

The color of the expression according to the type is shown in the table above. Color setting is possible in (Tools-Windows Attributes-Syntax Coloring (or Configure Current))

(- num1 num2)	*Calculates the value of num1 – num2*
(+ num1 num2)	*Calculates the value of num1 + num2*
(= num1 num2)	*Compares arguments for num1 and num2 equality*
(/= num1 num2)	*Compares arguments for num1 and num2 inequality*
(< num1 num2)	*Checks if num1<num2*
(> num1 num2)	*Checks T if num1 > num2*
(1+ num)	*Adds 1 in num*
(Abs num)	*The absolute value of num*
(Alert "c")	*Displays a window box containing an error or warning message about "c"*
(And A B)	*Returns T, if A, B are true.*
(Append list1 list2)	*Adds list 2 at list1.*
(Assoc item setlist)	*Outputs an element (list) that contains the specified item (item) from the setlist.*

(Cadr list1)	*Returns the second element of a list1*
(Car list1)	*Returns the first element of a list1*
(Cdr list1)	*Returns all the elements except the first element of a list1*
(Cond ((test-A) (A)) ((test-B) (B)))	*A function that accepts multiple functions as factors and performs formulas that meet each condition.*
(Defun Name)	*Defines a function which function name is Name. (Defines a sub-function)*
(Defun c:ARPAGO)	*Defines a function that command is ARAGO (Defines a main function)*
(Entget entity-name)	*Retrieves an entity-name's definition data*
(Equal A B)	*Verifies whether A and B is equal*
(Getreal "show command")	*"Show command" in the command window and receives a value of real number from the user.*
(Getkword "show command")	*Only a single string of the value specified in Initet is entered to the user.*
(Getvar "varname")	*Retrieves the value of an AutoCAD system variable*
(If (test-A) (then-A) (else-B))	*If (test-A) isn't nil, then-A works otherwise else-B executes.*
(Initget "keyword")	*Set the various options used by the Get** function.*
(Itoa num)	*The conversion of an integer into a string*
(Lambda (A) (+3 (/ A 2)))	*Define a zero function, similar to Defun. [A in examples defines A/2+3]*
(Last list1)	*Returns the last element in a list1*
(Length list1)	*Returns an integer indicating the number of elements in a list1*
(List A B C)	*Takes A,B,C and combines them into one list*
(Mapcar func list1)	*Executes the function on the individual elements of Lis1t.*
(Member A list1)	*If there is A in llist 1, return T. else nil.*
(Not A)	*Check whether the result of A is true or nil.*
(Nth num list1)	*Select the element that is located in the (num)th of the elements in List1.*
(Or A B C)	*Returns T if none of the elements A, B, C is nil.*

(Princ A File)	*Prints an A to the command line, or writes an expression to an file*
(Progn (A) (B) (C))	*Treat multiple expression and formulas in a single sentence*
(Repeat num A)	*Evaluates A (num) of times.*
(Setq A B)	*Save B to A.*
(Setvar "varname" A)	*Sets an AutoCAD system variable as A*
(SSget)	*Select one or more entities.*
(SSlength selection-set)	*Gets the number of entities in the selection-set.*
(SSname selection-set num)	*In selection-set, obtain the name of the (num)th drawing element.*
(Strcat A B C)	*Connect and respond to the listed A B C.*
(Subst A 'B list1)	*Replace B present in List1 with A.*
(Terpri)	*Prints a newline to the command line*
(Vl-load-com)	*Loads Visual LISP extensions to AutoLISP*
(Vl-registry-read "key")	*Returns data stored in the Windows registry for the specified key/value pair*
(Vl-remove A list1)	*If A is in list1, remove all of them.*
(Vl-sort list1 '<)	*Arrange the elements in the list in ascending order.*
(Vl-sort list1 '>)	*Arrange the elements in the list in descending order.*
(Vla-get-angle object)	*Obtain the angle of an object.*
(Vlax-curve-getpointatparam object 0.5)	*Obtain the vector to the center of the curve circle of the parameter when the object is curved.*
(Vlax-curve-getstartpoint object)	*Returns the start point of the object*
(Vlax-ename->vla-object name)	*Replace the name of the drawing element with the object.*
(Vlax-get-or-create-object "--.App")	*Returns the running --application object, or creates a new object.*
(Vlax-get-property A 'B)	*Obtains the B attribute of VLA object A.*
(Vlax-import-type-library	*Get the information from the library type.*

:tlb-filename FileName	*fileName means the library name, and the file type has an EXE TLB OLB TLB DLL.*
:methods-prefix "method"	*Specify the method prefix according to the file type (msxl- for Excel)*
:properties-prefix "property"	*Specify property prefix according to file type (msxl- for Excel)*
:constants-prefix "constant")	*Specify constant prefix according to file type (msxl- for Excel)*
(Vlax-invoke object1 'intersectwith object2 Extend-Option)	*Obtain the intersection of object1 and object2. In each case, Extend-Option indicates: [0=Not Extended, 1=Extended Select Object, 2=Extended Other Objects, and 3=Extended Both].*
(Vlax-invoke-method A 'B)	*The function that invokes the B method of object A*
(Vlax-put-property A 'B C)	*Set the B property of object A to C*
(Vlax-variant-value var)	*Transforms a specific value(var) into an object.*
(Zerop num)	*Verifies whether num is zero or not.*

Explanation of Syntax of “ARPA-GO Lisp”

- Main Function

```
(defun c:ARPAGO (/ oldsnap)
```

Define a command function called ARPA-GO. If “c:***” comes after defun function, *** is the command.

Declare local variable oldsnap. Make sure that there is a space between parentheses and commands to declare local variables

```
  (setq oldsnap(getvar "osmode"))
  (setvar "osmode" 0)
```

Save the currently set “osmode” to the variable oldsnap, and turn off all settings.

```
  (vl-load-com)
```

Import the Visual Lisp extension.

```
  (initget "N U")
  (setq mode(getkword
               "Choose Mode [ New Body(N) // Update Body(U) ]  :  "))(terpri)
```

The ‘New Body’ mode to extract coordinates of new hull form is set to “N”, and ‘Update body’ mode to extract coordinates of modified hull form is set to “U”. The letter should be entered by users and save it as variable mode.

```
(if (= mode nil)(setq mode "N"))
```

If mode=nil is true, save “N” in variable mode.

```
(if (= mode "N")(modeNB)(modeUB))
```

If mode=”N” is true, performs sub-function modeNB if not, sub-function modeUB.

```
(alert "Excel을 확인해주세요 !")

(setvar "osmode" oldsnap)
(princ)

);_메인함수 ARPAGO 종료
```

The notification window that says “Excel을 확인해주세요!” will come out and set osmode to oldsnap. If you do not write (princ) at the end of the function, nil is indicated in the execution window.

- Sub-Function

1. modeNB

```
(defun modeNB (/ wlspl num ni m objname jpoint Astlist Fstlist Alaylist
```

Define the sub functions 'modNB'. This function can be viewed as one big object of sub functions in order to extract coordinates when they are new drawing lines of ship.

To compare multiple Body plan in a one CAD file, you must use the New body mode several times. In order to avoid accumulation of data, all variables used in sub functions were declared as local variable (except sub functions 'setWL').

```
(setq halfB(getreal "Breadth/2 : "))(terpri)
```

This function receives the half Breadth of the ship and save it in the variable 'halfB'

```
(STapart)
```

Run the sub function 'STapart'.

```
(setq wlspl(ssget '((0 . "LINE,SPLINE,ARC"))))
```

Select only LINE, SPLINE, and ARC out of the multiple drawing elements selected by the user and save them to the variable 'wlspl'.

```
  (setWL)
  (setST)
  (offsetNB)
  (Xposition)
  (intersect)

  );_서브함수 modeNB 종료
```

After performing sub functions 'setWL, setST, offsetNB, Xposition, interconnect', the 'modeNB' function is finished.

2. modeUB

```
(defun modeUB()

  (offsetUB)
  (Xposition)
  (intersect)

  );_서브함수 modeUB 종료
```

A sub function 'modNB' is a union of sub functions in order to perform when the existing linear is modified. It already accepted the required variables through 'modNB' so, only performs sub functions 'offsetUB, Xposition, interconnect'.

3. STapart

```
(defun STapart()

  (setq stern(getreal "Stern Station Apart :  "))(terpri)
  (setq square(getreal "Square Station Apart  :  "))(terpri)
  (setq Bow(getreal "Bow Station Apart  : "))(terpri)

  );_서브함수 STapart 종료
```

The sub function 'STapart' is a function of receiving interval values for each part of the hull. Save the 'station' interval of the stern from the user to the variable 'stern', the 'station' interval of the main part to the variable 'square', and the 'station' interval of the bow to the variable 'Bow'.

4. setWL

```
(defun setWL()
```

Define the sub functions ‘setWL’. The function 'setWL' determines 'WaterLine' from the 'wlspl' list that only consists 'LINE, SPLINE, ARC' and save it as a list in the variable 'wllist'. This also obtains the ‘wlnumlist’, a list to represent the offset table according to the number of waterlines.

```
(setq wllist(list ) wlnumlist(list ))
```

Declare ‘Wllist’ and ‘wlnumlist’ as empty lists. This is the same reason why ‘modeNB’ declared as the region variable

'Wllist' is a list of the names of drawing plan of each waterline, and 'wlnumlist' is a list to indicate the type of offset table according to the number of waterline.

```
(setq num 0)
(repeat (sslength wlspl)
```

Save 0 in variable 'num' and repeat the sentence below until the set length of the 'wlspl' list.

```
  (setq name(ssname wlspl num))
  (if (and (= "LINE" (cdr(assoc 0 (entget name))))
           (or (zerop (vla-get-angle (vlax-ename->vla-object name)))
               (equal (vla-get-angle (vlax-ename->vla-object name)) pi 0.0001)))
    (setq wllist(append wllist (list name))))
  (setq num(+ num 1)))
```

Save the name of the drawing plan of set to the variable ‘name’ which is corresponding to the (num)th of the wlsplist.

The condition of the ‘if’ statement for waterline discrimination is when element of drawing plan is LINE, and angle should be 0 or 180 degrees', i.e. 'On the horizontal line'. The angle was obtained by the drawing name. This name of the drawing plan was changed to one object to obtain angle.

If the condition is true, 'if' statement is ended by adding the drawing plan name of the object to the 'wllist' list, 'repeat' will end after saving 'num+1' value to variable 'num'. Repeat this until the set length of the ‘wlspl’ list.

```
(setq wllist(vl-sort wllist '(lambda (x y)(< (cadr (cdr(assoc 10 (entget x))))
                                             (cadr (cdr(assoc 10 (entget y)))))))
```

Sort the wllist, which is the last list of drawing element names by each waterline, in ascending order based on the starting point coordinates (y).

```
(setq wllistn(length wllist))
```

To obtain 'wlnumlist', save the length of 'wllist' in variable 'wllistn'.

```
(setq wlnum 1)
(repeat (- wllistn 2)
  (setq wlnumlist(append wlnumlist(list wlnum)))
  (setq wlnum(+ wlnum 1)))

);_서브함수 setWL 종료
```

Save 1 in the variable 'wlnum' and then repeat the following syntax as much as 'wllistn-2'.

Add variable 'wlnum' to 'wlnumlist' and save it as variable 'wlnumlist'. Save the value 'wlnum+1' in variable 'wlnum'

5. separateST

```
(defun separateST()
```

Defines a sub-function separateST. To make users use modelUB, all station must be recognized in a different variable. ModelUB is the model to get the offset table of ship which the drawing lines have changed. SeparateST is a function that divides the stations into stern and bow parts of ship and save these in a list form.

```
(setq Astlist(list ) Fstlist(list ) Alaylist(list ) Flaylist(list ))
```

Just like setWL function which discussed before, users should start after clearing the list.

Astlsit : name of the station's drawing plan in stern part

Fstlist : name of the station's drawing plan in bow part

Alaylist : name of the station's layer in stern part

Flaylist : name of the station's layer in bow part

```
(setq num 0)
(repeat (sslength wlspl)
```

Save 0 in the variable num, and repeat the following syntax for the length of the wlspl set list.

```
(setq name(ssname wlspl num))
(setq layname(cdr (assoc 8 (entget name))))
(setq objname(vlax-ename->vla-object name))
```

Save the name of the drawing plan of the set to the variable 'name'. The set should be corresponds to the (num)th of the wlspl list. Save the layer name to variable layname and replace the name of the drawing plan with object and save it to the variable objname.

```
(if (= "SPLINE" (cdr(assoc 0 (entget name))))
  (progn
    (setq jpoint(car(vlax-curve-getpointatparam objname 0.5)))
    (if (< jpoint 0)
      (progn
        (setq Astlist(append Astlist(list name)))
        (if (= (member layname Alaylist) nil)
          (setq Alaylist(append Alaylist(list layname)))))
      (progn
        (setq Fstlist(append Fstlist(list name)))
        (if (= (member layname Flaylist) nil)
          (setq Flaylist(append Flaylist(list layname)))))))
```

In Body Plan, station is drawn as SPLINE or ARC. First of all, we will discuss when the type of elements is SPLINE.

If the type of element condition is true, use progn to compress it as one unity since there are many statements.

Save the X-coordinate of the center of the curve to the variable jpoint. If jpoint<0 is true, it is stern part station. Then it will save the name of drawing plan in the Astlist. And if the layer name is not in Alaylist, add layer name to Alaylist. If jpoint<0 is not true, the process is same as if it were true. But save in Fstlist and Flaylist instead of Astlist and Alaylist since it will be the station in bow part.

```
(progn
  (if (= "ARC" (cdr(assoc 0 (entget name))))
    (progn
      (setq jpoint(car(vlax-curve-getstartpoint objname)))
      (if (< jpoint 0)
        (progn
          (setq Astlist(append Astlist(list name)))
          (if (= (member layname Astlist) nil)
            (setq Alaylist(append Alaylist(list layname)))))))))
```

Also in the case of the type of element condition is false, use progn to compress it as one unity since there are many statements.

Give the condition one more time to check whether the type of element is ARC or not. Save X-coordinates of the start point to jpoint. Save in Astlist, Fstlist depending on whether the jpoint condition is met, same as in the case of SPLINE.

```
(setq num(+ num 1)))
```

Save the value of num+1 in the variable 'num' before the start of repeat statements, and then close the repeat statements. Repeat the syntax up to here by the length of the wlspl set list. And then differentiate the station of stern and bow part through checking the condition of every element users have chosen.

The criteria for layer names are as follows and it is recommended to use the same name as possible since the station number indicated on Excel is also based on this. ST T2, ST T1, ST A.P. , ST 0.5, ST 1, ST 1.5, ST 2, ST 3, ST 4, ST 5, ST 6, ST 7, ST 8, ST 9, ST 10, ST 11, ST 12, ST 13, ST 14, ST 15, ST 16, ST 17, ST 18, ST 18.5, ST 19, ST 19.5, ST F.P. , ST F1, ST F2. .

```
(setq Alaylist(vl-sort Alaylist '<))
```

Sort Alaylist in ascending order to extract coordinates from Transom.

```
(setq Alaylistn(length Alaylist))
```

The type of ST T1, ST T2, ST A.P. is string and is aligned behind. To correct this, save the length of the Alaylist to the variable Alaylistn.

```
(setq Alastlist(append Alastlist (list (nth (- Alaylistn 1) Alaylist)
                                       (nth (- Alaylistn 2) Alaylist)
                                       (nth (- Alaylistn 3) Alaylist))))
```

Add the (Alaylistn-1), (Alaylistn-2), and (Alaylistn-3)th list of the Alaylist to the variable ‘Alastlist’.

```
(setq Alaylist(vl-remove (nth 0 Alastlist) Alaylist))
(setq Alaylist(vl-remove (nth 1 Alastlist) Alaylist))
(setq Alaylist(vl-remove (nth 2 Alastlist) Alaylist))
```

Clear the 0th, 1st and 2nd element of the Alastlist from Alaylist. The Alaylist remained 0.5 to 10 stations.

```
(setq Alaylist(append Alastlist Alaylist))
```

Adding the Alaylist to the Alastlist, the Alaylist is marked after the element of Alastlist. Save the created list as the variable ‘Alaylist’. This list is arranged in order from Transom.

```
(setq Flaylist(vl-sort Flaylist '>))
(setq Flaylistn(length Flaylist))

);_서브함수 separateST 종료
```

Flaylist is bow part’s station, ST 10 must be at the beginning of this list and ST F2 must be at the end of the list. Therefore, Flaylist is in descending order and then save the length of the Flaylist in variable ‘Flaylistn’ to save in different variables by each station.

6. setAST

```
(defun setAST()
```

Define the sub-function setAST. This function is used to separate stern stations with different variables using Alaylist and Astlist. If the lines of drawing plan are the same, the layers are also grouped together. This is why we separated the stern stations based on the length of the layer name list Alaylist and declared the layer names from 10 to 15. The variables which consist the stern station are sta, stb, stc, std, ste, stf, stg, sth, sti, stj, stk, stl, stm, stn, sto from Transom to 10 station. This sub-function repeats the same syntax several times, therefore only part of it has been excerpted.

```
  (if (= Alaylistn 15)
    (progn
      (setq nn 0)
      (repeat (length Astlist)
```

If Alaylistn = 15, that is, the layer name exists from Transom to ST10. If the condition is true, start with prong since there are multiple executable statements. Save 0 in variable ‘nn’ and repeat the following syntax for Astlist(number of stern drawings names).

```
        (cond

          ((= (cdr(assoc 8 (entget(nth nn Astlist)))) (nth 0 Alaylist))
           (setq sta (append sta (list (nth nn Astlist)))))
```

The conditional statement ‘cond doesn’t have specific statement of execution and goes on to the next condition in the case of false condition. If the last condition is false, exit the cond statement. In this case, you get the layer name of the nth station element stored in the astlist, determine if it matches the nth layer name in the Alaylist list and save it in the variable according to station.

If ‘layer name of the nnth element of Astlist = the 0th element of Alaylist is true, Add the nnth element (drawing name) of the Astlist to the sta list. If the condition is false, move on to the next condition. Since the length of the Alaylist is 15, the condition declares up to the 14th element of the Alaylist and closes the cond statement after that.

```
        (setq nn(+ nn 1)))))
```

Close all repeat, prong, and if statements after saving the value of nn + 1 in variable ’nn’.

```
    (setq #sta sta #stb stb #stc stc #std std #ste ste #stf stf #stg stg
        #sth sth #sti sti #stj stj #stk stk #stl stl #stm stm #stn stn #sto sto)
  );_서브함수 setAST 종료
```

In order to find the intersection point, sta, stb, etc. which is saved value by each station must maintain the stored value, but since the element is appended to the list, you must declare it as a region variable to prevent the list element from accumulating when reusing the function. Therefore, by storing each value in another variable, it is possible to obtain the intersection by declaring sta etc. as a local variable.

7. setFST

```
(defun setFST()
  (if (= Flaylistn 14)
```

Define the sub-function setFST. It is a function for separating the bow part stations into different variables by using Flaylist and Fstlist, and the procedure is the same as setAST. Since ST 10 is distinguished from the stern section, we have declared 10 layers to 14 layers. The variables stored in the bow part station are staa, stbb, stcc, stdd, stee, stff, stgg, sthh, stii, stjj, stkk, stll, stmm, stnn in that order from ST F2 to ST 9.

```
  (setq #staa staa #stbb stbb #stcc stcc #stdd stdd
        #stee stee #stff stff #stgg stgg
        #sthh sthh #stii stii #stjj stjj #stkk stkk
        #stll stll #stmm stmm #stnn stnn)
  );_서브함수 setAST 종료
```

As with setAST, variables are stored as new variables in order to declare local variables, which are the forms of adding elements

8. intersect

```
(defun intersect()
```

Define the sub-function 'interside'. This function finds the intersection point of the 'waterline' list and 'station' arranged in ascending order and extracts coordinates with Excel. When you look at Body Plan, there is a station drawn as one object or two objects, this executes a sub-function according to each condition. Since the sentences for each station are the same, we are going to discuss about the Transom part for example.

```
(if (= (length #sta) 1)
  (progn
    (setq obj2(vlax-ename->vla-object (nth 0 #sta)))
    (caseone)
    (setq mm 2)
    (inputExcel))
  (progn
    (setq obj3(vlax-ename->vla-object (nth 0 #sta)))
    (setq obj4(vlax-ename->vla-object (nth 1 #sta)))
    (casetwo)
    (setq mm 2)
    (inputExcel)))
```

If the condition 'when there is only one object making a Transom' is true, '#sta' is in the form of a list. It should take the name of drawing plan located at zero and replace it with the object in the variable 'obj2'. Run the sub function 'caseone' to find the intersection point of the waterline list. In the offset table, Transom locates in the 2nd row. We can save the value as 2 in the variable 'mm' and run sub function 'inputExcel' to enter the values in Excel.

```
(if (= (length #staa) 1)
  (progn
    (setq obj2(vlax-ename->vla-object (nth 0 #staa)))
    (caseone)
    (setq mm 30)
    (inputExcel))
  (progn
    (setq obj3(vlax-ename->vla-object (nth 0 #staa)))
    (setq obj4(vlax-ename->vla-object (nth 1 #staa)))
    (casetwo)
    (setq mm 30)
    (inputExcel)))

);_서브함수 intersect 종료
```

After performing this process in every station, the sub function 'intersect' exits.

9. caseone

```
(defun caseone()

  (setq blist(list ))
```

Define the sub function 'caseone'. This is a function to find the intersection point of the waterline list and station when a station is drawn with one object in the 'interside' function. 'blist' is the list of the x coordinates of the intersection to be obtained through this function, and 'caseone' is used multiple times within the 'intersect' function. So we have to make sure that the list is empty in time to start. This is to avoid accumulating data.

```
(setq ni 0 m 0)
(repeat wllistn
```

Save 0 in variables 'ni' and 'm' respectively as initial values and repeat the following statements by the 'wllistn' (waterline list length).'

```
  (setq wl(nth ni wllist))
  (setq obj1(vlax-ename->vla-object wl))
  (setq interp(vlax-invoke obj1 'intersectwith obj2 0))
  (setq selectx(car interp))
```

Save the 'ni'th number of the wllist in variable 'wl' and change wl which is the name of drawing plan to the object obj1. Obtain the intersection point of the specified obj2 in the obj1 and intersection function, save it as a variable interp, and save only the first element, the x coordinate, as the variable selectx.

```
  (if (/= selectx nil)
    (progn
      (setq selectx(abs selectx))
      (setq blist(append blist(list selectx))))
    (setq blist(append blist(list m))))
  (setq ni(+ ni 1)))
```

If the condition 'selectx is not nil' is true, it means that the intersection point of the station and its corresponding draft exists, and it is added to the 'blist' by placing an absolute value on the x coordinates.

If the condition is false, add '0' to 'blist' because the intersection does not exist. If this process is not performed, the Excel 'offset table' will show 'nil' instead of '0'.

Exit the repeat statement after saving the value ni +1 in the variables 'ni'. If you repeat these statements by the amount of 'wllistn', you will get a 'blist', a list of intersection points with the 'station' and 'waterline' lists.

```
(if (= (last blist) 0)
  (setq blist(subst halfB '0 blist)))

);_서브함수 caseone 종료
```

However, the 'ST 10' typically has the same value as the half breadth of ship in certain draft, but the intersection of 'waterline' and 'station' does not exist in the drawing, resulting in a 'nil' value. To solve this problem, use the fact that the last 'waterline' value at any location in the 'station' has value.

If the condition 'when the last element of the blist is 0' is true, replace the '0' present in the 'blist' with 'halfB' (half breadth of ship entered by the user in the beginning). This determines the 'blist' for that 'station'.

10. casetwo

```
(defun casetwo()
  (setq clist(list ) dlist(list ) blist(list ))
```

Define the sub-function casetwo. In the intersect function, if the station is drawn with two objects, it is actually a function that finds the intersection point of the waterline list and the station. clist is the the list to find the intersection point of the waterline list with the first element making the station, and dlist is the list of intersection with the second element. A blist is the list of the intersection point of the stations to be finally obtained using the two. Like caseone, empty the lists to start. The process of obtaining the intersection points and saving points in clist and dlist is the same as caseone.

```
  (setq blist (mapcar '(lambda (x y) (+ x y)) clist dlist))
  );_서브함수 casetwo 종료
```

Add each element of clist and dlist together to get the blist as a result value.

11. LoadExcel

```
(defun LoadExcel ()
```

Define the sub-function LoadExcel. This function connects AutoLisp and Excel to send offset values to Excel.

```
  (if (and (setq excelPath (vl-registry-read
                            "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\Cu
                            "Path"))
           (setq excelPath (strcat excelPath "Excel.exe")))
```

Save the path of executable file of Excel to excelPath.

```
    (progn
      (if (not msxl-acos)
        (vlax-import-type-library
          :tlb-filename excelPath
          :methods-prefix "msxl-"
          :properties-prefix "msxl-"
          :constants-prefix "msxl-"))
      (setq ExcelApp (vlax-get-or-create-object "Excel.Application")))
```

Retrieve the method, properties, and information from excelPath and return or generate an Excel application object and save it in the variable ExcelApp.

```
ExcelApp
```

```
);LoadExcel 종료
```

Run ExcelApp and exit the function LoadExcel.

12. offsetNB

```
(defun offsetNB()
```

Define the sub-function offsetNB. When user selects modeNB, it is a function to specify the format of offset which will appear in Excel, and can be divided into the display format of offset values, the display format of waterline number, and the display format of station number and actual x-coordinate of station.

```
(setq ExcelApp (LoadExcel))
(vlax-put Excelapp "visible" :vlax-false)
(setq Workbooks (vlax-get-property ExcelApp 'Workbooks))
(setq CurrentWBook (vlax-invoke-method Workbooks 'Add))
(setq Sheets (vlax-get-property ExcelApp 'Sheets))
(setq AcSheet (vlax-get-property ExcelApp 'ActiveSheet))
(setq Cells (vlax-get-property AcSheet 'Cells))
```

The sub-function LoadExcel is a function that connects to Excel and executes it. Save this process to the variable ExcelApp. If vlax-true, it will run Excel on the screen. We declared this value as false since running Excel automatically may cause discomfort to the user. Save the open Excel file to the variable worksbooks and create a new Excel file and save it to the variable CurrentWBook to write the data to the Excel by getting intersection points. The current sheets are saved in the variable Sheets and the currently active sheet is saved in the variable AcSheet. Save the cells of the active sheet to the variable Cells.

```
(setq Ran (vlax-get-property Excelapp 'Range "B2:Z100"))
(vlax-put-property Ran 'Numberformat "0.000")
```

Since the display format of the offset value is rounded to the fourth decimal place, we specified the format of the number. Save the properties of the cells after B2 in the variable Ran. Use the Numberformat to specify the number format, up to the third decimal place.

```
(setq colY 1 rowX 3)
(setq val "BOTTOM")
(vlax-put-property Cells 'Item colY rowX val)

(setq colY 1 rowX 4)
(setq val "0.5 WL")
(vlax-put-property Cells 'Item colY rowX val)

(setq colY 1 rowX 5)
(setq wlnumlist(mapcar '(lambda(x)(strcat (itoa x) " W.L"))wlnumlist))
(mapcar
  '(lambda (val)
     (vlax-put-property Cells 'Item colY rowX val)
     (setq rowX(1+ rowX)))wlnumlist)
```

These command lines are the display format of the waterline number. First, enter the name of the waterline horizontally, starting from BOTTOM, in cell C1, which is the row 1 and column 3. Replace the wlnumlist elements created by sub-function setWL with a letter and input “W.L” to enter it into the cell.

```
(setq wx 3)
(repeat wllistn
```

This is the part of the display format of the waterline number that set the properties of the cell. Start with column C1, so save 3 to variable wx and repeat the number of waterlines.

```
  (setq colY 1 rowX wx)
  (setq cel(vlax-variant-value(vlax-get-property Cells 'Item colY rowX)))
```

To get the properties of the cell, hold the cell as an object and save it in the variable cel.

```
  (setq int(vlax-get-property cel 'Interior))
  (setq Font(vlax-get-property cel 'Font))
  (setq borders(vlax-get-property cel 'Borders))
```

To set the background color of the cell, get the Interior property and store it in the variable int. To set the thickness of the font, get Font property and store it in the variable Font. Get the Borders property for formatting the borders and stores it in the variable borders.

```
  (setq bordersB(vlax-get-property borders 'Item 9))
  (setq bordersR(vlax-get-property borders 'Item 10))
```

Get the property at the bottom of the border and save it in the variable bordersB. Get the property of the rightmost part and save it in the variable bordersR. If you don’t get these properties, you can’t partly set the thickness, type, etc. of the border.

```
    (vlax-put-property cel 'HorizontalAlignment -4108)
    (vlax-put-property int 'Color 14599344)
    (vlax-put-property Font 'Bold "true")
    (vlax-put-property bordersB 'Weight 4)
    (vlax-put-property bordersB 'LineStyle 1)
    (vlax-put-property bordersR 'Weight 2)
    (vlax-put-property bordersR 'LineStyle 1)
    (setq wx(+ wx 1)))
```

Arrange to the center the horizontal alignment of the values entered in the cell. Change the background color and set the font thickness to bold. The bottom border is a bold continuous style border, with the right edge set to a thin continuous style border, and then ends the repeat statement by storing wx+1 in the variable wx.

```
  (setq Ran(vlax-get-property Excelapp 'Range "A1:B1"))
  (vlax-put-property Ran 'MergeCells "true")
```

Since the station number and the actual x-coordinate form of the station are mostly the same as the WL display format, We will discuss about the different matters only. As you can see from the format of the offset table extracted through ARPA-GO, A1 cell and B1 cell of Excel file are merged and marked “No.station”. To do this, we set A1:B1 to the variable Ran, and merge cells using Mergecells.

```
  (setq bordersV(vlax-get-property borders 'Item 11))
  (setq bordersH(vlax-get-property borders 'Item 12))
```

Since it is a cell specified as a range rather than a cell, it obtains the property of the vertical border of the internal border and saves it in the variable bordersV, and the property of the horizontal border is saved in variable bordersH.

13. offsetUB

The sub-fuction offsetUB is a function for specifying the type of offset table to appear in Excel when the user selects modeUB. Most of things are the same as the offsetNB function, but in this we use the list created by the setWL function and offsetNB function without recreating the lists such wlnumlist, backlist, etc.

14. InputExcel

```
(defun InputExcel()
```

Define the sub-function InputExcel. It is a function that sends the final coordinate lease to Excel.

The coordinate lease obtained from the intersect function which obtains the intersection point.

```
(setq xlist blist)
(setq colY mm rowX 3)
(mapcar
  '(lambda(val)
     (vlax-put-property Cells 'Item colY rowX val)
     (setq rowX (1+ rowX)))xlist)

);_서브함수 InputExcel종료
```

Save blist, a list of x coordinates obtained by the intersect function, as variable xlist. The value of the row is variable mm, specified in the intersect function for each station. Insert xlist horizontally from the cell.

15. Xposition

```
(defun Xposition()
```

Define the sub-function Xposition. It is a function to input the actual x-coordinate into Excel using station spacing of stern part, main part, and bow part inputted through STapart. Calculate the position by considering the number of stations for each part and enter it in Excel. Since the same or previously mentioned phrases are used, only the stern part has been excerpted.

```
(setq colY 2 rowX 2)
(setq val (- 0 (* 2 stern)))
(vlax-put-property Cells 'Item colY rowX val)
```

Since the value to enter in row2, column2 (that is, B2) is the x coordinate of Transom, after doubling the stern value, save the value subtracted from 0 in variable ‘val’. In cell row2 and column2, enter val.

16. setST

```
(defun setST()

  (separateST)
  (setAST)
  (setFST)

  );_서브함수 setST 종료
```

Define the sub-function setST. This function is a bundle of sub functions used to distinguish all stations.

4.1.2 Class

Python has built-in support for Object Oriented Programming (OOP). It is an interpreted, high-level, general-purpose programming language.

Classes are the core of Python. Python language defines class which is the basic unit of object-oriented programming as one bundle. Not only class, even the program which contains various variables defines as a bundle.

Most beginners and even some advanced Python programmers do not understand the distinction between instance and class variables. Their lack of understanding forces them to use these different types of variables incorrectly.

The basic difference is following:

1) Instance variables are for data which is unique to every object
2) Class variables are for data shared between different instances of a class

To use class method, the class reserved word, class name, and colon are used on the first line of coding. Then we have to define parameter which is internal function of Class and call the parameter to derive the results. It is same as a regular function, except that it is also part of the class.

Notice

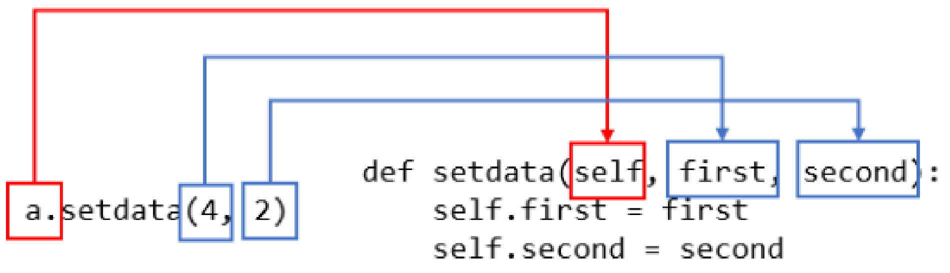
It is customary to follow Pascal notation when naming a class.

Class call method

1) Method to call parameter through object.

   You create an object named 'a' and call the parameter 'setdata' through created object 'a'.

   This 'setdata' requires three parameters but the results show only two values, (4,2). This is because the first parameter 'self' is connected to the object 'a' which calls parameter 'setdata'.



```
a.setdata(4, 2)
```

```
def setdata(self, first, second):
    self.first = first
    self.second = second
```

Notice

To call a method of a class through an object, you must use the dot(.) Operator, as in a.setdata(4,2).

---

2) Method to call parameter through Class

The case we call parameter as a form of "class name. Parameter", object 'a' must be connected to 'self', the first parameter.

This method is not popular. Most of people use first method we introduced.

```
>>> a = FourCal()
>>> FourCal.setdata(a, 4, 2)
```

Class Characteristics

1) Class defines the structure and behavior of the object.
2) Classes of objects are controlled through initialization.
3) Class makes complex problems easy.

4.2 Terms

To calculate ship, users need to have knowledge about terms of ship. In this chapter, we will discuss about the terms ARPA-GO used.

- LOA (Length overall): Length overall (LOA) is the maximum length of a vessel's hull measured parallel to the waterline. It is the most commonly used way of expressing the size of a ship.
- LWL (Length on the Waterline) : A vessel's waterline length (LWL) is the length of a ship at the level where it sits in the water
- LBP (Length Between Peripherals) =LPP : Length between perpendiculars (LBP) is the length of a ship along the waterline from the forward surface of the stem, or main bow perpendicular member, to the after surface of the sternpost, or main stern perpendicular member. When there is no sternpost, the centerline axis of the rudder stock is used as the aft end of the length between perpendiculars
- B : Breadth of the ship
- D: Depth of the ship.
- Displacements : weight of water that a ship pushes aside when it is floating, which in turn is the weight of a ship
- KB: KB is the center of buoyancy which is the height above the keel.
- OB: the distance from the center of the buoyancy to the waterline.
- LCB (Longitudinal Center of Buoyancy): The centroid of the underwater volume of the ship expressed as a longitudinal location. This centroid is connected with stability of ship
- LCF: Longitudinal Center of Flotation (LCF) is the center of the water plane.
- BM: BM is the metacentric radius of ship.

(BML: radius of longitudinal meta-center, BMT: radius of transverse meta-center)

- KM : Height from the keel to meta-center (KB+BM)
- TPC (Tons per cm): Tons per centimeter (TPC), i.e. the number of tons required to sink the vessel one centimeter.
- MTC(Moment to change Trim one cm): Longitudinal moment required for 1cm trimming
- KML: the distance from the keel to the longitudinal metacenter
- Cb: The block coefficient of a ship is the ratio of the underwater volume of ship to the volume of a rectangular block having the same overall length, breadth and depth.
- Aw: water plane area.
- Cw: waterline coefficient, It is the ratio of the actual area of the water plane to the product of the length and breadth of the ship.
- Cm : The midship section coefficient (Cm) is the ratio of the area of the immersed midship section (Am) at a particular draft to that of a rectangle of the same draft and breadth as the ship
- Cp : The prismatic coefficient of a ship at any draft is the ratio of the volume of displacement at that draft to the volume of a prism having the same length as the ship and the same cross-sectional area as the ship's midships area..
- Cvp : vertical prismatic coefficient, A large value of vertical prismatic coefficient indicates will indicate body sections of U-form and a low will indicate V-sections.