



Volume XXIV 2021

ISSUE no.2

MBNA Publishing House Constanta 2021



Scientific Bulletin of Naval Academy

SBNA PAPER • OPEN ACCESS

Standalone application for hull volume calculation based on Simpson's rule using PYQT5

To cite this article: A. PINTILIE, P. BURLACU, E. G. ROBE-VOINEA, G. C. CRISTEA, Scientific Bulletin of Naval Academy, Vol. XXIV 2021, pg. 128-134.

Submitted: 02.12.2021

Revised: 02.02.2022

Accepted: 15.03.2022

Available online at www.anmb.ro

ISSN: 2392-8956; ISSN-L: 1454-864X

doi: 10.21279/1454-864X-21-I2-012

SBNA© 2021. This work is licensed under the CC BY-NC-SA 4.0 License

STANDALONE APPLICATION FOR HULL VOLUME CALCULATION BASED ON SIMPSON'S RULE USING PYQT5

Alexandru PINTILIE¹, Paul BURLACU², Elena-Grațîela ROBE-VOINEA³
Georgiana-Cătălina CRISTEA⁴

¹Ovidius University of Constanța, Romania, ²Naval Academy of Constanța, România,
³Naval Academy of Constanța, România, ⁴Naval Academy of Constanța, România

elena.robe@anmb.ro

Abstract.

The area and volume estimation for an irregular surface is a real challenge, especially when it comes to a ship. This paper aims to give an insight into the process of developing a software for hull volume approximation based on Simpson's Rule, using PyQt5. It comprises a brief introduction, information about Simpson's integrating method, along with its usage in the hull volume approximation, the developing process of the software and a testing section.

Key words: Hull volume, Simpson's integration, Python, programming.

1. Introduction

To begin with, it is very important to understand the mathematical package that implies the rules to tackle with this issue. In our case we choose as a state of art the Simpson's Rule as the best way to approximate the ship's area and volume values. In the second part, this paper deals with the computational implementation of the above mentions by creating a user-friendly interface. In this respect, it was necessary initially to develop the algorithm, then to implement it, and finally bring it to the standalone level. To make this feasible, we used an interpreted high-level general-purpose programming language, Python, along with its PyQt5 special library, for both programming and graphic user interface aspects.

2. Simpson's Rule

When it comes to integrating, there are no general rules as there are for differentiating for example. Also, even if most of the mathematical functions can be differentiated analytically, integration cannot be. For instance, let's consider the function e^{-x^2} . Its indefinite integral cannot be found analytically. Therefore, we need numerical methods for evaluating integrals. This relies on the fact that the definite integral of a function $f(x)$ between the limits $x = a$ and $x = b$ equals the area under $f(x)$ bounded by the x-axis and the two vertical lines mentioned above. In other words, numerical methods

comprise various methods for estimating the area under $f(x)$. One way to achieve this is by using Simpson's Rule.

2.1. Simpson's Rule from a Mathematical point of view

To start with, Simpson's Rule is a numerical integration method, which is more accurate than the Trapezoidal Rule. Practically, it divides the area under the function to be integrated, $f(x)$, into vertical strips. However, instead of connecting the points $f(x_i)$ with straight lines, every group of three such subsequent points is fitted with a parabola.[1] To ensure that there are always an even number of panels, the step-length h is normally chosen so that there are $2n$ panels, i.e., $n = (b - a)/(2h)$. Adopting the notation above, Simpson's rule estimates the integral as:

$$S = \frac{h}{3} [f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(x_{2i}) + 4 \sum_{i=1}^n f(x_{2i-1})] \quad (1)$$

2.2 Simpson's Rule from the hull volume calculation point of view

The concept for this issue is again represented by dividing the geometry into distinct smaller-sized pieces, calculating the volume of each such piece, and adding them up together through integration, thus, approximating the final result, the volume of the entire geometry. Nonetheless, the section of a hull differs along the length of the ship. With regard to this matter, we use the Simpson's Rule, as it is also suitable for the more complicated forward and aft ends of the vessel, as a consequence of its accuracy. With reference to the previous subsection of the paper, and considering to have an odd number of sections of the ship, the hull volume calculation formula for the first three sections, based on Simpson's Rule would be:

$$V = \frac{h}{3} (A_0 + 4A_1 + A_2) \quad (2),$$

as shown in the figure below:

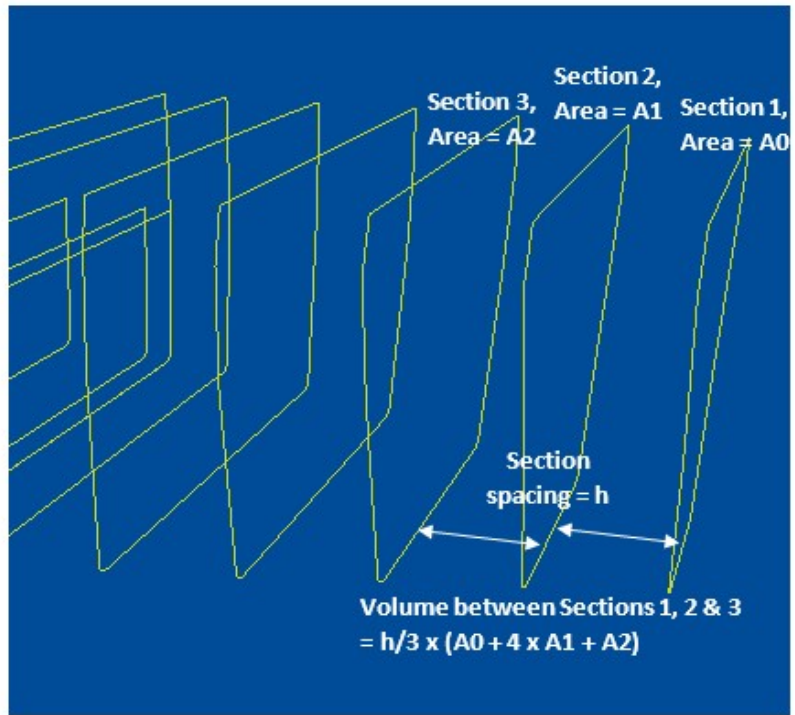


Figure 1[2]

Given the fact that every odd-numbered section has a multiplier of 4, every even-numbered section has a multiplier of two, and the two end sections have no multipliers. We can adapt the above formula for several sections as shown below:

$$V = \frac{h}{3}(A_0 + 4A_1 + 2A_2 + 4A_3 + 2A_4 + 4A_5 + \dots + A_n)(3)$$

Furthermore, if we want to calculate the sections' areas as well, we can use the same concept as for the volume. Using the notations from Figure 2 as a reference, the formula for every section's area would be:

$$A = \frac{c.i.}{3}(y_1 + 4y_2 + 2y_3 + 4y_4 + y_5)(4)$$

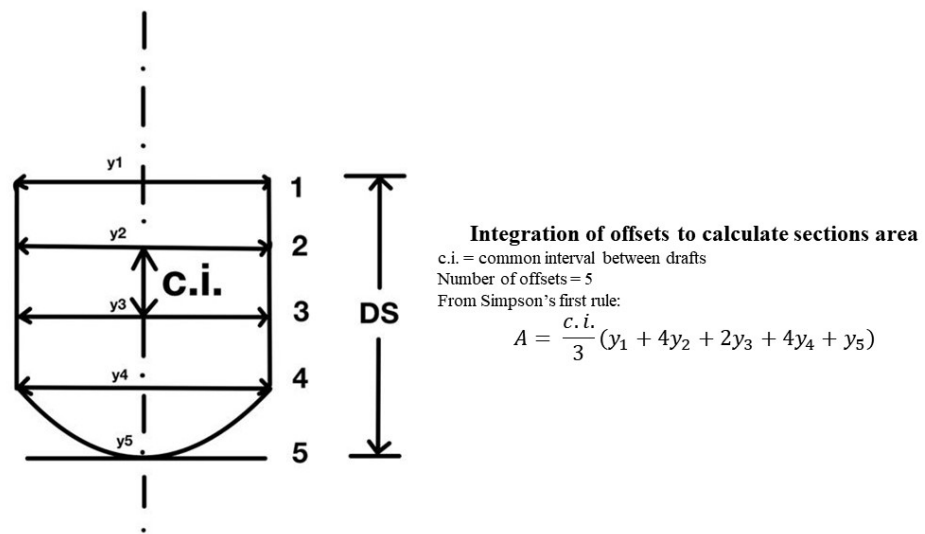


Figure 2

3. The Computational Implementation

For the computational implementation of the concept presented earlier in this paper, we chose to use Python as a base. Furthermore, we utilised two different softwares, QtDesigner for the visual designcomponent, and PyCharm Community, along with PyQt5 for the programming part. Finally, we brought the application to the standalone level, HullVolumeCalculator.exe, using PyInstaller.

“PyQt consists of a number of Python bindings for cross-platform applications that combine all the strengths of Qt and Python. Using PyQt, you can include all Qt libraries in Python code, so you can write GUI applications in Python. In other words, you can use PyQt to access all the facilities provided by Qt through Python code.”[3]

3.1. The visual designcomponent

As mentioned above, for the design of the application it was used the software named QtDesigner(Figure 3). “Qt provides a great application for setting up the layouts and designing main windows, widgets, or dialogs. Qt Designer is a graphical interface filled with Qt widgets and other tools used for building GUIs. Using the Qt Designer application’s drag and drop interface, you are able to create and customize your own dialogs, windows, and widgets.”[4]

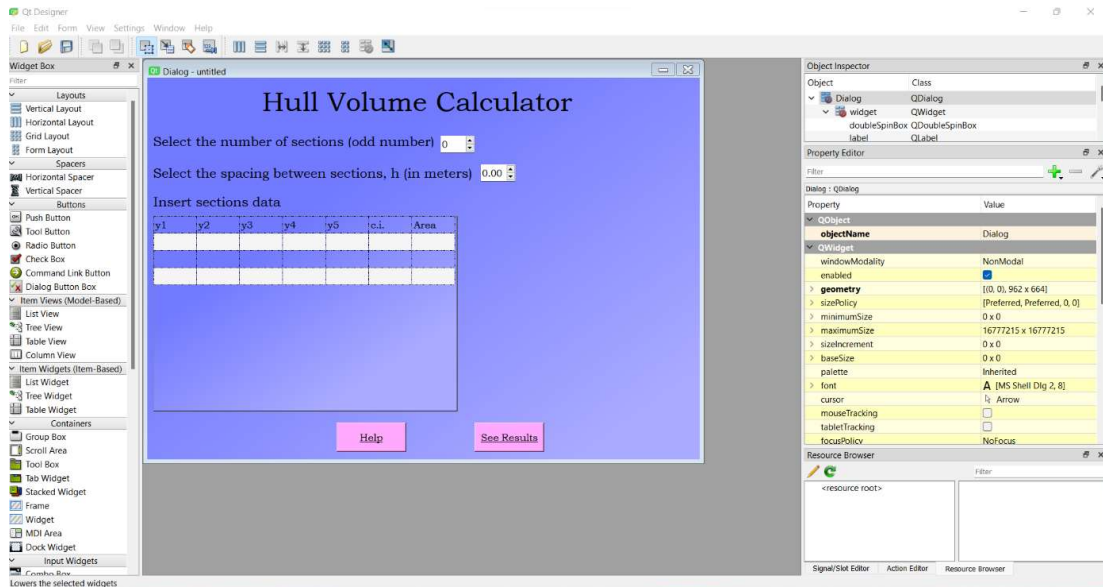


Figure 3

For our application we designed two distinct windows, a main window and a help window, as shown in the figures below.

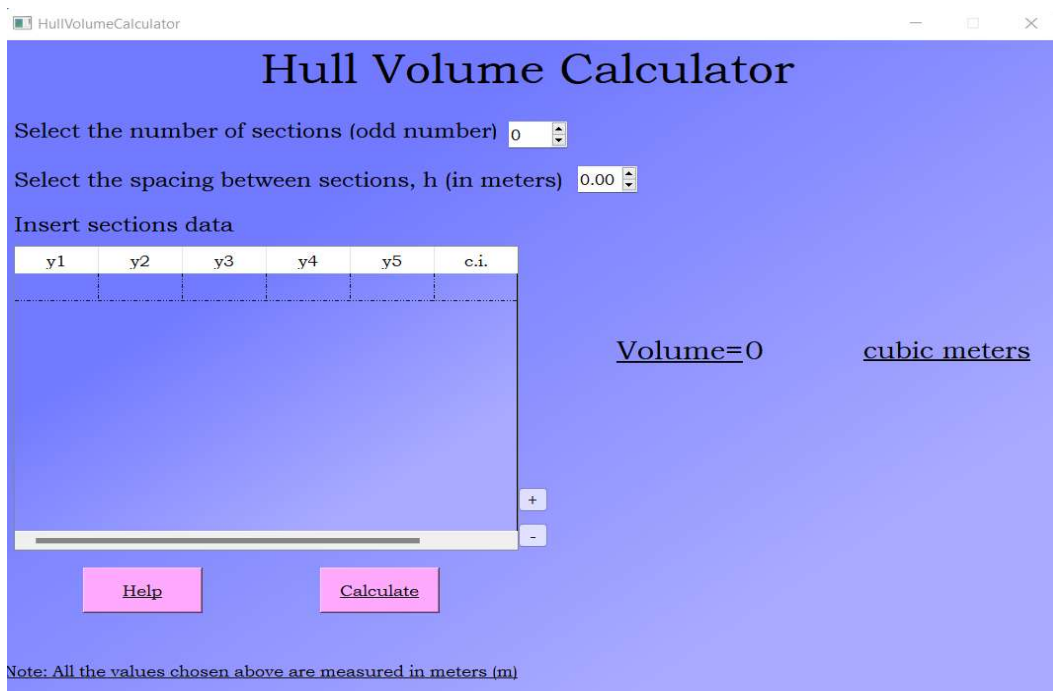


Figure 4

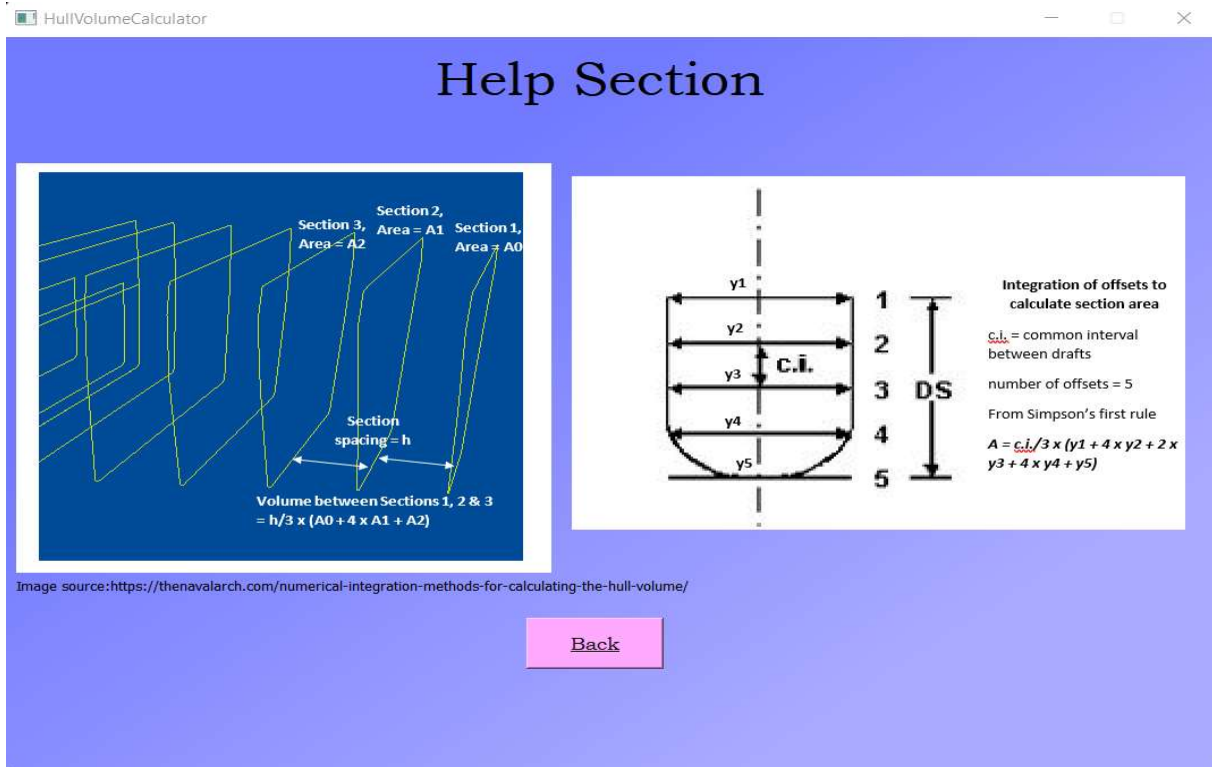


Figure 5

The main window (Figure 4) of the application comprises a variety of widgets such as labels, used for writing text, spinboxes, used for choosing different values, buttons for accessing the application's functions and a table for storing the ship's section's values. Its purpose is to assess the user into calculating the ship's hull volume.

The help window (Figure 5) is composed of labels, two pictures and a back button, all aiming to offer the user a better understanding of the previous window and also the possibility to go back to it.

3.2. The programming part

This section of the application was achieved using PyCharm and PyQt5, as mentioned above.

PyCharm Community is a free software provided by JetBrains that ensures the user the possibility to write and read python algorithms. In regard of everything mentioned in the second part of the paper, we created a function for the hull volume calculation:

```

4. def calcul(self):
rows = self.table.rowCount()
row = 0
areas = []
for row in range (nrows):
y1 = self.table.item(row, 0)
y2 = self.table.item(row, 1)
y3 = self.table.item(row, 2)
y4 = self.table.item(row, 3)
y5 = self.table.item(row, 4)
ci = self.table.item(row, 5)
y1 = float(y1.text())
y2 = float(y2.text())
y3 = float(y3.text())

```

```

y4 = float(y4.text())
y5 = float(y5.text())
ci = float(ci.text())
area = (ci/3)*(y1+4*y2+2*y3+4*y4+y5)
areas.append(area)
area1 = areas[0]
arean = areas[-1]
sum = area1 + arean
i=1
for i in range (1, (len(areas)-1)):
if i % 2 == 0:
    sum = sum + 2*areas[i]
else:
    sum = sum + 4*areas[i]
h = self.hvalue.value()
volume = (h/3) * sum
self.resultlabel.setText(str(format(volume, ".4f")))

```

However, the whole programming of the application includes much more, such as importing different libraries or distinct functions that insure the software's interactivity with the user, and it can be found in the main.py file.

3.3. The conversion of the .py file to a standalone application (.exe)

This final step was achieved by using PyInstaller. "PyInstaller freezes (packages) Python applications into standalone executables, under Windows, GNU/Linux, Mac OS X, FreeBSD, Solaris and AIX." [5] Moreover, dividing this action into two steps, the first was installing this function from PyPI by writing a line a Command Window(CMD), and the second was represented by running another command in the directory where the .py file was.

4. Testing HullVolumeCalculator.exe

In order to test our application, we used ship data from a body plan found in [6], for a chemical tanker (140000Dwt). After that, we adapted the extracted data to meet our application's requirements.

The obtained results are shown in the figure below. Finally, the application estimated the ship's volume without technical errors.

Select the number of sections (odd number) 21

Select the spacing between sections, h (in meters) 13.7

Insert sections data

y1	y2	y3	y4	y5	c.i.
33.35	0.01	0	0	0	5.1
40.38	22.01	0.49	2.98	0	5.1
44.79	40.96	14.39	8.47	0	5.1
45.91	45.17	39.57	20.43	6.75	5.1
45.98	45.85	45.31	38.86	25.05	5.1
46	46	46	45.65	37.14	5.1
46	46	46	45.96	38.27	5.1
46	46	46	45.96	38.27	5.1
46	46	46	45.96	38.27	5.1
46	46	46	45.96	38.27	5.1

Volume=211884.cubic meters

Help Calculate

Note: All the values chosen above are measured in meters (m)

Figure 6

5. Conclusions

The standalone application for hull volume calculation based on Simpson's Rule, using Python language, was born from the concept presented in the article "Numerical integration methods for hull volume estimation" [10], and it represents an improvement and an upgrade of both the graphic design and sustainability of the program. Furthermore, the source code keeps the proposed mathematical calculation algorithm, without generating errors.

In the future, the application could be upgraded to a new version, where it can fit other integration methods such as trapezoidal rule, that can be used for calculating the hull volume, as well. A newer version could also comprise the possibility to choose the number of offsets to be used in the estimation.

References

- [1] Valentine, Daniel T., and Brian Hahn. *Essential MATLAB for engineers and scientists*. Academic Press, 2019, 305-306
- [2] Team TheNavalArch, *Numerical integration methods for hull volume estimation*, Maritime Industry, Naval Architecture, Ship Design and Construction, January 5, 2021, <https://thenavalarch.com/numerical-integration-methods-for-calculating-the-hull-volume/>
- [3] Siahaan, Vivian, and Rismon Hasiholan Sianipar. *LEARNING PyQt5: A Step by Step Tutorial to Develop MySQL-Based Applications*. Sparta publishing, 2019, 1
- [4] Willman, Joshua M. "Creating GUIs with Qt Designer." *Beginning PyQt*. Apress, Berkeley, CA, 2020. 165-203.
- [5] PyInstaller Development Team, *PyInstaller*, August 11, 2021, <http://www.pyinstaller.org/>
- [6] Dragomir, Dumitru, *Ship's shapes album*, Editura Fundației Universitare "Dunărea de Jos", Galați, 2008