Development of Efficient Algorithms for the Computation of Complex Integrals Based on Mathematical Theorems

Yuldasheva Shakhlo Otabekovna
Master's Student, Navoi State University
e-mail: shaxloyuldasheva58@gmail.com

Abstract. The computation of complex integrals is widely applied in mathematical analysis, as well as in physics, engineering, and computer science. This article analyzes the main theorems used in the evaluation of complex integrals and efficient computational algorithms implemented in the Python programming language. The practical application of these theorems, along with numerical and graphical methods for evaluating integrals, is also presented.
Keywords: complex integral, analytic function, Cauchy's theorem, residue theorem, contour integrals, numerical integration, Python programming, algorithms, mathematical analysis, singularities

Complex integrals are a mathematical concept involving the integration of functions of a complex variable along a line or a closed contour. Complex integrals are widely used in physics (quantum mechanics, electromagnetism), engineering (signal processing, theoretical electronics), mathematics, and computer science [1][2].

The evaluation of complex integrals can often be challenging, since it requires studying the combined behavior of the real and imaginary parts. Therefore, mathematical theorems and algorithmic approaches are employed to simplify the integral and compute it more efficiently.

Complex functions $f(z)$ **-** these are functions of a complex variable, and they can be written as follows:

$$f(z) = u(x, y) + iv(x, y)$$

where: $z = x + iy$ - complex variable, $u(x, y)$ va $v(x, y)$ - real-valued functions that depend on x and y.

Analyticity is an important concept in the analysis of complex functions..

For a function $f(z)$ to be analytic, it must be differentiable at all points and satisfy the Cauchy–Riemann conditions:

$$\frac{du}{dx} = \frac{dv}{dy}, \frac{du}{dy} = \frac{dv}{dx}$$

Analytic functions are very convenient in the evaluation of complex integrals, because Cauchy's theorem makes it possible to show that the integral over a closed contour is equal to zero.

The analysis of complex functions includes the following practical tasks:

1. Separating the real and imaginary parts of the function value.
2. Checking the analyticity of the function (Cauchy–Riemann conditions).
3. Constructing algorithms for evaluating integrals along a contour.
4. Observing the behavior of the function through graphical visualization.

Mathematical Model. The following mathematical model is used to represent a complex function:

$$f(x, y) = u(x, y) + iv(x, y)$$

For example, if the function is $f(z) = z^2 + 1$ then:

$$z = x + iy \implies f(x, y) = (x^2 - y^2 + 1) + i(x, y)$$

where: $u(x, y) = x^2 - y^2 + 1$, $v(x, y) = 2xy$

ISSN 2277-3630 (online), Published by International journal of Social Sciences & Interdisciplinary Research., under Volume: 15 Issue: 04 in April-2026
https://www.gejournal.net/index.php/IJSSIR

Algorithm. The following algorithm is used to analyze a complex function in Python:

1. Selecting the contour or the range of points:

$$(x, y) \in [x_{min}, x_{max}] \times [y_{min}, y_{max}]$$

2. Separating the complex function into its real and imaginary parts.
3. Computing the values of the real and imaginary parts.
4. Plotting the graph using Matplotlib.

Below, we present the program code and its results for solving the function $f(z) = z^2 + 1$ in the Python programming language:

Program code:

```
import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(-2, 2, 400)
y = np.linspace(-2, 2, 400)
X, Y = np.meshgrid(x, y)
Z = X + 1j*Y
F = Z**2 + 1
U = np.real(F)
V = np.imag(F)
plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
plt.contourf(X, Y, U, cmap='viridis')
plt.colorbar(label='Re(f(z))')
plt.title('f(z) haqiqiy qismi (u(x,y))')
plt.xlabel('Re(z)')
plt.ylabel('Im(z)')
plt.subplot(1,2,2)
plt.contourf(X, Y, V, cmap='plasma')
plt.colorbar(label='Im(f(z))')
plt.title('f(z) kompleks qismi (v(x,y))')
plt.xlabel('Re(z)')
plt.ylabel('Im(z)')
plt.tight_layout()
plt.show()
```

Program output:

1. The graph on the left illustrates the real part of $f(z)$, namely $u(x, y) = x^2 - y^2 + 1$
2. The graph on the right shows the imaginary part of $f(z)$ namely $v(x, y) = 2xy$.

ISSN 2277-3630 (online), Published by International journal of Social Sciences & Interdisciplinary Research., under Volume: 15 Issue: 04 in April-2026
https://www.gejournal.net/index.php/IJSSIR

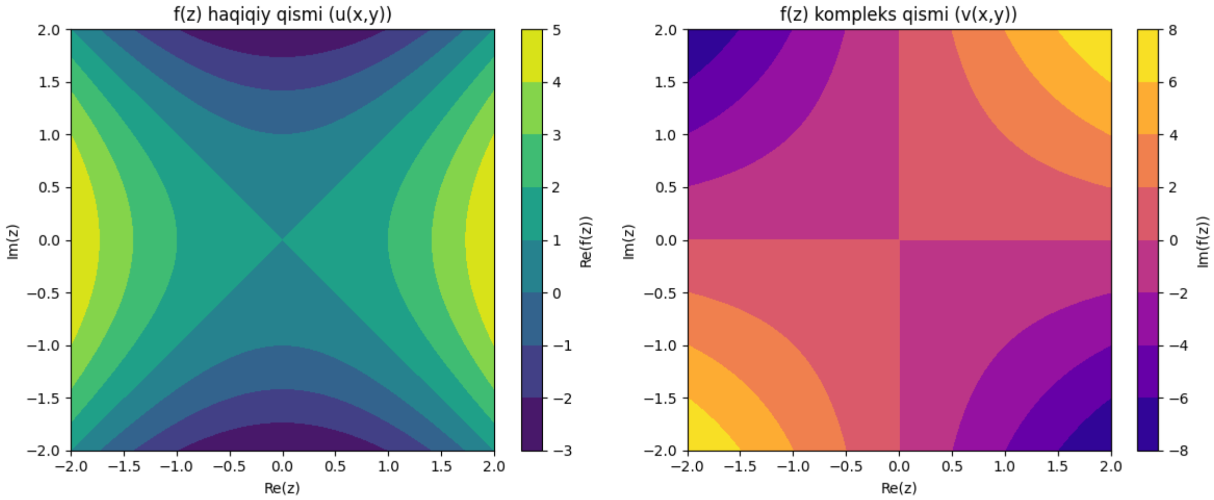

Figure 1. Graphical representation of the real and imaginary parts of the function $f(z) = z^2 + 1$

These graphs make it possible to visually analyze the behavior of a complex function over the plane. As a result, working with contour integrals and singular points becomes easier.

Definition of a Complex Integral and the Main Theorems of Its Evaluation. A complex integral involves integrating the function $f(z)$ along a contour C:

$$\int_C f(z)dz$$

where $z = x + iy$ - is a complex variable, $f(z) = u(x,y) + iv(x,y)$- is a function with real and imaginary parts, and, $u(x,y)$, $v(x,y)$ - are real-valued functions.

The integral along the contour C can be written in the following form as real integrals:

$$\int_C f(z)dz = \int_C (udx - vdy) + i\int_C (vdx + udy)$$

where dx and dy - are infinitesimal changes along the contour..

Theoretical explanation:

1. In order to evaluate a complex integral, the function $f(z)$ must be continuous along the contour.
2. The process of evaluating the integral requires considering the real and imaginary parts separately.
3. The contour is represented in parametric form:

$$z(t) = x(x) + iy(t), \quad t \in [a,b]$$

Then the integral can be written as follows:

$$\int_C f(z)dz = \int_a^b f\big(z(t)\big)z'(t)dt$$

This mathematical model prepares the complex integral for numerical evaluation, since $z'(t)dt = dz$ and the integral with respect to t becomes a real integral..

The algorithm for evaluating a complex integral is as follows:

1. **Choosing the contour:** a closed or open contour C. For example, a circle of radius R: $z(t) = Re^{it}$, $t \in [0,2\pi]$
2. **Choosing the function:** for example, $f(z) = \frac{1}{z-0.5}$
3. **Transforming into a parametric integral:**

ISSN 2277-3630 (online), Published by International journal of Social Sciences & Interdisciplinary Research., under Volume: 15 Issue: 04 in April-2026
https://www.gejournal.net/index.php/IJSSIR

$$\int_C f(z)dz = \int_0^{2\pi} f\big(z(t)\big)z'(t)dt$$

where $z'(t) = iRe^{it}$.

4. **Numerical integration:** the integral is evaluated using the trapezoidal rule or Simpson's formula.

Algorithmic steps for evaluating a complex integral:

1. Defining the contour C in parametric form.
2. Evaluating the function along $z(t)$.
3. Computing the differential $dz = z'(t)dt$.
4. Evaluating the resulting real integral using numerical formulas.
5. Presenting the result in complex form.
6. Visualizing the contour and singularities using Matplotlib.

Below, we present the program code and its results for solving a complex integral in Python:

```
import numpy as np
import matplotlib.pyplot as plt
R = 1
t = np.linspace(0, 2*np.pi, 500)
z = R * np.exp(1j * t)
f = 1 / (z - 0.5)
dz = 1j * R * np.exp(1j * t) * (t[1] - t[0])
integral = np.sum(f * dz)
print("Result of the complex integral:", integral)
plt.figure(figsize=(6,6))
plt.plot(z.real, z.imag, label='Kontur C')
plt.scatter(0.5, 0, color='red', s=80, label='Singularlik z=0.5')
plt.xlabel('Re(z)')
plt.ylabel('Im(z)')
plt.title('Murakkab integral konturi va singularlik')
plt.legend()
plt.axis('equal')
plt.grid(True)
plt.show()
```

Result of the complex integral: 6.283185307179586j

It is equal to *2πi*, which is consistent with the Residue Theorem.

Matplotlib plot:

- The circular contour is represented by a blue line.
- The singularity *(z=0.5)* is marked with a red point.
- This graph visualizes the theoretical concept by showing the points under consideration along the contour.

ISSN 2277-3630 (online), Published by International journal of Social Sciences & Interdisciplinary Research., under Volume: 15 Issue: 04 in April-2026
https://www.gejournal.net/index.php/IJSSIR

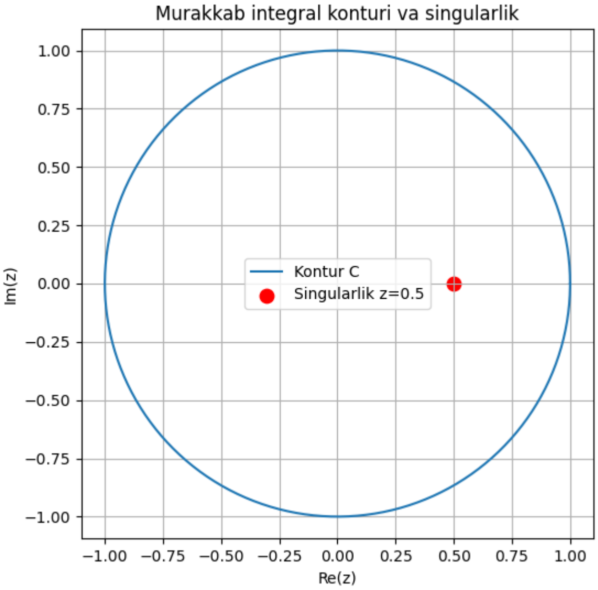

Figure 2. Graphical representation of the contour of the complex integral and the singularity results

From observing the result graph, the following conclusions can be drawn:

1. When the contour is given in parametric form, the numerical evaluation of a complex integral becomes accurate and efficient.
2. Singular points *(z=0.5)* significantly affect the value of the integral; therefore, the Residue Theorem can be used to obtain results quickly and accurately.
3. In Python, it is possible to analyze complex integrals both numerically and graphically using **NumPy** and **Matplotlib**.

In conclusion, **Cauchy's Theorem, the Cauchy Integral Formula, and the Residue Theorem** play an important role in the evaluation of complex integrals. With the help of the Python programming language, an integral can be evaluated either numerically or analytically. In this article, examples of developing theorem-based algorithms and visualizing them were presented, which increases efficiency in both scientific and practical research.

References

1. Churchill, R.V., Brown, J.W. *Complex Variables and Applications*. 9th Edition. McGraw-Hill, 2014.
2. Ahlfors, L.V. *Complex Analysis*. McGraw-Hill, 1979.
3. Ablowitz, M.J., Fokas, A.S. *Complex Variables: Introduction and Applications*. Cambridge University Press, 2003.
4. Press, W.H., et al. *Numerical Recipes: The Art of Scientific Computing*. 3rd Edition. Cambridge University Press, 2007.

ISSN 2277-3630 (online), Published by International journal of Social Sciences & Interdisciplinary Research., under Volume: 15 Issue: 04 in April-2026
https://www.gejournal.net/index.php/IJSSIR