# Project Idea for GSoC 2018

# SciPy

# Complex Analysis

## General Info

Required knowledge and skills: Python, understanding of all the concepts of complex analysis like properties of complex numbers, functions with complex numbers as arguments, continuity, analyticity, harmonicity, integration and contour integration of complex functions.

Difficulty level: Medium

Mentors: Not available till date

## Description

Complex Analysis is an important branch of mathematics and is used in various fields of science and technology. It deals with complex numbers and functions with complex numbers as arguments. Any complex number is denoted by *z* and the associated function by *f(z)*. The definition of various terms related to complex numbers is presented below:

1. **Complex Numbers**: Any ordered pair of real numbers, (x,y) represented as x+*j*y, where *j*=√-1. In short, *z*=x+*j*y
2. **Functions with complex numbers as arguments**: Any function with complex number as arguments is an ordered pair of functions, u(x,y) and v(x,y) represented as u(x,y)+*j*v(x,y). In short, f(z)=u(x,y)+*j*v(x,y).

The definitions of continuity, analyticity, harmonicity, integration and contour integration for complex functions can be seen at the links mentioned in the references.

The main aim of this project will be to create a module which will allow to check the continuity, differentiability, analyticity, and harmonicity of complex functions at various points. This module will also allow to perform differentiation, integration and contour integration of complex functions. The proposed module name is **scipy.math.complex**. The name can be changed after discussion with mentors.

The main class of the module will be **ComplexAnalysis**.

The **ComplexAnalysis** will be having the following methods:-

1. __*iscont*__: This method will check the continuity of complex functions at the points passed as arguments.
2. __*isanalytic*__: This method will check the analyticity of complex functions at the points passed as arguments.
3. __*isharmonic*__: This method will check the harmonicity of complex functions in the finite complex plane and return the list of singular points which makes the function non-harmonic.
4. __*isdiff*__: This method will check the differentiability of complex functions at the points passed as arguments.
5. __*diff*__: This method will return the complex derivative at the points passed as arguments.
6. __*integrate*__: This method will integrate the real and imaginary parts of complex functions passed as arguments and will return the value obtained in, complex representation.
7. __*cintgerate*__: This method will find the integration of complex function along the complex curve passed as arguments.

Apart from the methods we can use the following theorems and properties for faster computation:

1. Cauchy Integral Theorem
2. Cauchy Integral Formula
3. Properties of complex polynomials, that is, *P(z)* and *Q(z)* where *z* is a complex number.

ComplexAnalysis should have the property that it should be able to convert one representation of complex numbers to another. For that we can use inbuilt methods or we can make our own methods after discussion with mentors.

## References:

1. http://home.iitk.ac.in/~gp/MSO202Lect2.pdf
2. http://home.iitk.ac.in/~gp/MSO202Lect3.pdf

If any mentor is interested in mentoring me as a student in GSoC 2018 then I would be keen to develop this idea of Complex Analysis further in detail.

Feedbacks are positively welcomed.

**Gagandeep Singh**

**IIT Jodhpur, Karwad**

**Rajasthan**

**India**

**Pin Code: 342037**