A Computer Program For The System Of Weak Fuzzy Complex Numbers And Their Arithmetic Operations Using Python

Lama Razouk [1,*], Suliman Mahmoud [2], Mohammad Ali [3]

[1,2,3]Department of Mathematics, Faculty of Sciences, Tishreen University, Latakia, Syria

Emails: Lamarazouk94@gmail.com; Suliman_mmn@yahoo.com; MohamadAli@gmail.com

[*]Corresponding Author: Lamarazouk94@gmail.com

Abstract

Weak fuzzy complex number system is considered as a generalization of complex numbers by using an algebraic element with fuzzy property. It is believed that this new numerical system will play many important roles in various computer science disciplines, proceeding from this importance this paper aims to introduce for the first time the algebraic operations defined on weak fuzzy complex numbers using Python and Jupyter Notebook. Hence, by building a new *class* and *its functions* in Python for this new set, we can now use weak fuzzy complex numbers and the main arithmetic operations on them.

Keywords: weak fuzzy complex number; split-complex numbers; Python; Jupyter Notebook

1. Introduction:

According to the continuing need to develop, extending sets of numbers is improved over time. Starting with natural numbers $\boldsymbol{N}$, integers $\boldsymbol{Z}$, rational $\boldsymbol{Q}$ then real numbers $\boldsymbol{R}$ which was first discovered by German Mathematician Julius Wilhelm Richard Dedekind.[2]

After that, the novel of the generalization of real numbers is told us by several sets, like:

- Complex number $\boldsymbol{C} = \{\boldsymbol{x + j\,y; j^2 = -1}, \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{R}\}$
- Dual numbers $\boldsymbol{D} = \{\boldsymbol{x + yj}; \boldsymbol{j^2 = 0}, \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{R}\}$
- Split- complex numbers $\boldsymbol{S} = \{\boldsymbol{x + yj}; \boldsymbol{j^2 = 1}, \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{R}\}$.

We mention that weak fuzzy complex numbers and split-complex numbers were used on a wide range in the study of vector spaces, Diophantine equations, and matrices, see [4-7].

In parallel, computer programming plays an important role in providing us with the most ideal options for accomplishing tasks, especially with enormous calculations on those sets. Somehow, the above-defined sets were used in many programming languages before [3].

Noteworthy, Python is one of the most high-level programming languages. It supports three numeric types to represent numbers: integers(*int*), float(*float*), and complex(*complex*) numbers. More precisely, ready complex functions are using the module *math* and useful tools to handle and manipulate them. For example, Python converts the real numbers *x* and *y* into complex using the function *complex*(*x*,*y*). The real part can be accessed using the function *real* () and the imaginary part can be represented by *imag*e (). This tuple (real, imag) seems the same for numbers in Dual and split-complex sets and for numbers in the following set too.

At the beginning of this year 2023, a new set of numbers was defined as follows:

- $\boldsymbol{F_j = \{a + bj\ ;\ a, b \in R\ ,\ t = j^2 \in]0, 1[\}}$, the set of *weak fuzzy complex numbers,*

where *'j'* is the weak fuzzy complex operator ($\boldsymbol{j \notin R}$) [1]. We will mention its properties in section 2.

Nevertheless, the numbers in $\boldsymbol{C, D, R,}$ and $\boldsymbol{F_j}$ represent tuples (*x*,*y*) similarly, the value of the operator *'j'* is different between them *x*+*yj*. So we can determine the kind of number by defining its operator like in our

Doi: https://doi.org/10.54216/GJMSA.080104
Received: March 25, 2023 Revised: June 23, 2023 Accepted: September 26, 2023

implementation *check_operator( )*. We will call the operator *'k'* to avoid confusion with *'j'* which is known as the complex operator in Python.

In section 3, we will define the numbers of set $\boldsymbol{F_k}$ and their operations by the class *WFCNumber* and its functions. Also, we will add some examples.

Properties of the weak fuzzy complex numbers:[1]

Let $x = a + bk$, y = c + d$k \in F_k$,

- Addition: $x + y = (a + c) + (b + d)\,k$.
- Multiplication $x.\ y = (a + bk).\ (c + dk) = (ac + bdt) + (ad + bc)\,k$.
- The conjugate of x is: $\bar{x} = a - bk$
- The norm of x is defined as follows: $\|x\| = \sqrt{|x.\bar{x}|} = \sqrt{|a^2 - b^2 t|}$.
- x is invertible if and only if $t \neq \frac{a^2}{b^2}$ and $\frac{1}{x} = x^{-1} = \frac{a}{a^2 - tb^2} + k\frac{-b}{a^2 - tb^2}$.

1. The implementation:

In our work, we have checked the type of operator '*k*' by the function *check_operator*( ). The plan below shows the process.

Also, the primary computing on weak fuzzy complex numbers is done by ***class*** *WFCNumber*.

Implementation for checking the operator 'k':

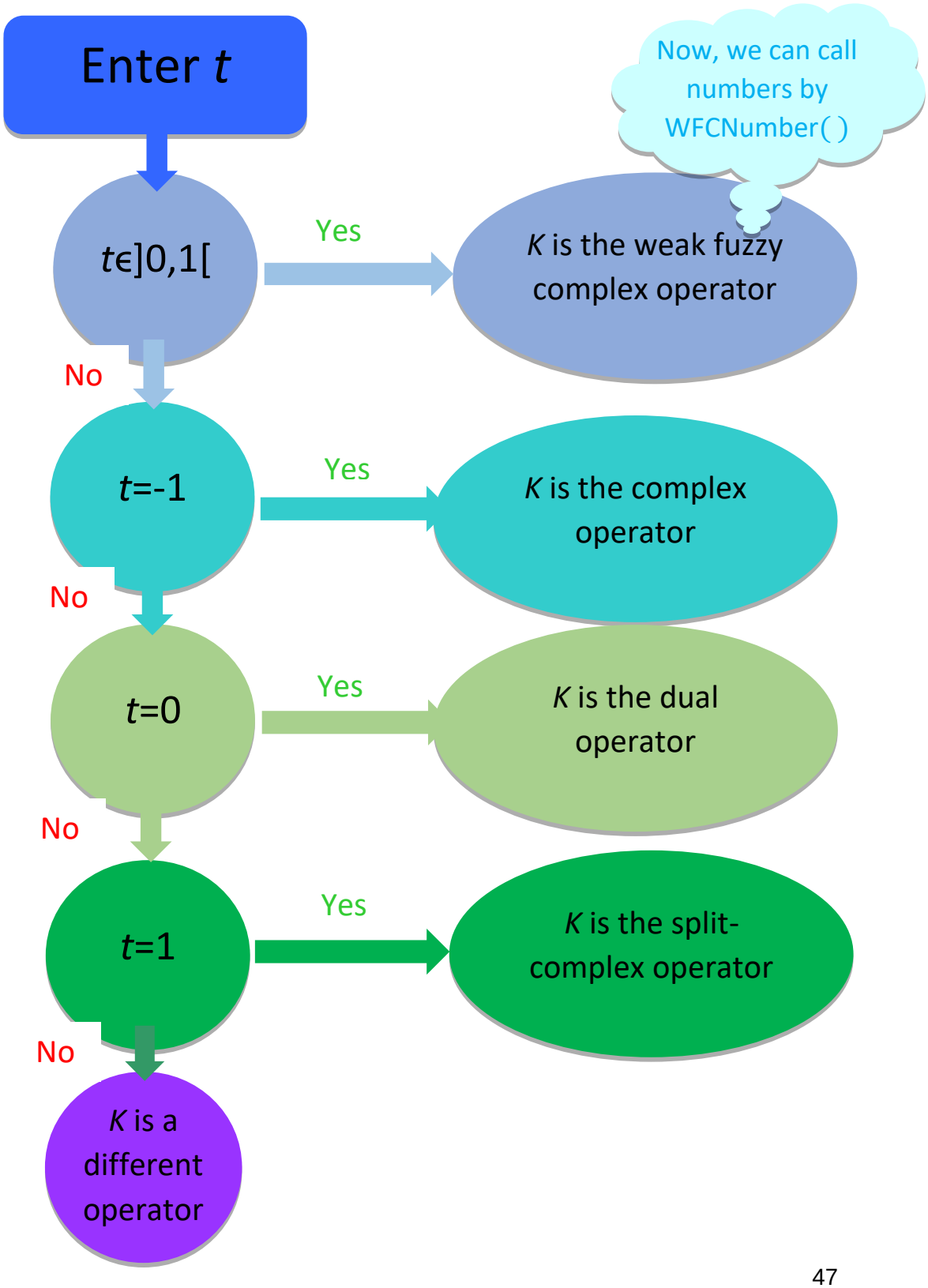
To distinguish the operator '*k*'(complex, dual, split-complex, or weak fuzzy complex), the function *check_operator*( ) does the task where $t=k^2$.

```
import math

import cmath

import pandas as pd


def check_operator (t):
   I=pd.Interval (left=0.0, right=1.0, closed='neither')
   if t in I:
      k=math.sqrt(t)
      print( "k=" , k)
      print("k is a weak fuzzy complex operator")
       print("Enter the related weak fuzzy complex numbers like:  name_number=  WFCNumber  (real,imag,t)")
   elif t==-1:
      k=cmath.sqrt(t)
      print( "k=" , k)
      print("k is the complex imaginary operator")
   elif t==0:
      k=math.sqrt(t)
      print("k=" , k)
      print( "k is the dual imaginary operator" )
   elif t==1:
```


Doi: https://doi.org/10.54216/GJMSA.080104
Received: March 25, 2023 Revised: June 23, 2023 Accepted: September 26, 2023

```
    k=math.sqrt(t)
    print( "k=" , k)
    print("k is the split-complex imaginary operator")
else: print("k is not a complex imaginary operator nor split nor fuzzy")
return
```

check_operator( ) function



Doi: https://doi.org/10.54216/GJMSA.080104
Received: March 25, 2023 Revised: June 23, 2023 Accepted: September 26, 2023

Examples of use *check_operator*( ) **function:**

1) check_operator(0.4)

output:
k= 0.6324555320336759
k is a weak fuzzy complex operator
Enter the related weak fuzzy complex numbers like: name_number=WFCNumber(real,imag,t)

2) check_operator(0)

output:
k= 0.0
k is the dual imaginary operator

3.2 The main definition and operations Implementation :

Inside the body of *class WFCNumber*( ), we have defined fuzzy complex numbers as triple **(real part, imaginary part, k^2)**.

Our class includes the following functions:

function	operation
__add__	Addition
__sub__	Subtraction
__mul__	Multiplication
__truediv__	Division
__conj__	Conjugate
__norm__	Norm
__invert__	Invertible
the_type	Number type

```
class WFCNumber:
    def __init__(self, real, imag, t):
        self.real = real
        self.imag = imag
        self.t = t

    def __repr__(self):
```


Doi: https://doi.org/10.54216/GJMSA.080104
Received: March 25, 2023 Revised: June 23, 2023 Accepted: September 26, 2023

```
        return "{}{:+}k".format(self.real, self.imag, self.t)

    def __add__(self, other):
        real = self.real + other.real
        imag = self.real + other.imag
        return WFCNumber(real, imag, self.t)

    def __sub__(self, other):
        real = self.real - other.real
        imag = self.real - other.imag
        return WFCNumber(real, imag, self.t)



    def __mul__(self, other):
        real = self.real * other.real + (self.imag * other.imag* self.t)
        imag = self.real * other.imag + self.imag * other.real
        return WFCNumber(real, imag, self.t)

    def __truediv__(self, other):
        if self.t!=other.real**2/other.imag**2:
            denom = other.real**2 - self.t * other.imag**2
            real = (self.real * other.real - self.t * self.imag * other.imag) / denom
            imag = (self.imag * other.real + self.real * other.imag) / denom
            return WFCNumber(real, imag, self.t)
        else:
            return ' not possible'

    def __conj__(self):
        real = self.real
        imag = -self.imag
        return WFCNumber(real, imag, self.t)



    def __norm__(self):
        aa=self.real
        bb=self.imag
```


Doi: https://doi.org/10.54216/GJMSA.080104
Received: March 25, 2023 Revised: June 23, 2023 Accepted: September 26, 2023

```
        tt= self.t
        return math.sqrt(abs(aa**2 - tt*bb**2))


    def __invert__(self):
        if self.t!=self.real**2/self.imag**2:
            denom = self.real**2 - self.t * self.imag**2
            real = self.real / denom
            imag = -self.imag  / denom
            return WFCNumber(real, imag, self.t)
        else:
            return ' not possible'



    def the_type (self):
        if type(self).__name__=='WFCNumber':
            return 'Weak Fuzzy Complex'
        else:
            print(type(self).__name__)
            return


#Enter numbers(examples)

import math

a=WFCNumber(1, 2, 0.5)
b=WFCNumber(2, 3, 0.5)

#operators for 2 numbers
print("Addition is :" , a+b)
print("Subtraction is :" , a-b)
print("Multiplication is :" , a * b)
print("Division is :" , a / b)

#for one number:
print("The conjugate of " , a ," is :" , a.__conj__())
print("The norm of" , a , "is: " , a.__norm__())
print("The invertible of " , a , "is:",a.__invert__())
print("The type of" , a , "is: " , a.the_type())
```

Output:
Addition is : 3+4k
Subtraction is : -1-2k
Multiplication is : 5.0+7k
Division is : 2.0-14.0k
The conjugate of 1+2k is : 1-2k
The norm of 1+2k is: 1.0
The invertible of 1+2k is: -1.0+2.0k

Doi: https://doi.org/10.54216/GJMSA.080104
Received: March 25, 2023 Revised: June 23, 2023 Accepted: September 26, 2023

The type of 1+2k is: Weak Fuzzy Complex

5. Conclusion

In this paper, we have used Python and Jupyter Notebook to define the *class* of weak fuzzy complex numbers for the first time and present functions for their properties such as norm, conjugate, and invertible for each number. And addition, subtraction, multiplication, division for two numbers, and more.

In the future, our objective is to make algorithms to solve linear and quadratic weak fuzzy complex equations. Also, to define weak fuzzy complex matrices in Python.

Reference

[1] Hatip, A., "An Introduction To Weak Fuzzy Complex Numbers ", Galoitica Journal Of Mathematical Structures and Applications, Vol.3, 2023.

[2]Kumbhar, A. and Shinde, A., "REAL NUMBERS AND THEIR INTERESTING PROOFS ",

JBNB: A Multidisciplinary Journal, Vol. 12, 2022.

[3] Gutin, R., "Matrix decompositions over the double numbers", arXiv:2105.08047v2,

https://doi.org/10.48550/arXiv.2105.08047, 6 Dec 2021.

[4] Khaldi, A., " A Study On Split-Complex Vector Spaces", Neoma Journal Of Mathematics and Computer Science, 2023.

[5] Ahmad, K., " On Some Split-Complex Diophantine Equations", Neoma Journal Of Mathematics and Computer Science, 2023.

[6] Ali, R., " On The Weak Fuzzy Complex Inner Products On Weak Fuzzy Complex Vector Spaces", Neoma Journal Of Mathematics and Computer Science, 2023.

[7] Hatip, A., "On The Weak Fuzzy Complex Vector Spaces", Galoitica Journal Of Mathematical

Structures And Applications, Vol.3, 2023.

Doi: https://doi.org/10.54216/GJMSA.080104
Received: March 25, 2023 Revised: June 23, 2023 Accepted: September 26, 2023