Vectorial_Library

Release 16/07/2020

CABOS Matthieu

Dec 22, 2021

CONTENTS:

Here is my new version of the Vectorial library.

This library is an optimized vectorial computation api. Made with Cython, it works as fastest as a standard C++ or C librairy. This module have been splitted into 2 main parts :

- **The Point Object** : It allows to manipulate Point Object into a 3 Dimension environnement. The main operators have been implemented. This object are compatible with the following Vector Class. The avaible methods are listed as follow :
  - **copy** : Clone the current object
  - **op** : Operator facrtorization code implementing all standard operator (+,-, ...)
  - **print_screen** : Standard terminal scrren printer
  - **Getters and Setters**
- **The Vector Object** : It permit to manipulate Vector Objects into a 3 dimensions environnement. This class use Point as parameters, many of these methods are the "well known" vector standard manipulation operators. The avaible methods are listed as follow :
  - **angle** Get the alpha angle between vector and norm.
  - **angle_vec** Get the alpha angle between two vectors
  - **copy** Get a clone of the current object
  - **cos** Get a cosinus from two vector's angle
  - **dot** Compute standard dot between two vectors
  - **hortogo** Assert hortogonality of two vectors
  - **length** Get the vector's **real** length
  - **op** Operator facrtorization code implementing all standard operator (+,-,*, ...)
  - **pointAt** Compute the translation from given Point and Vector
  - **print_screen** Standard terminal scrren printer
  - **prod** Compute standard cross product between two vectors
  - **setLength** Set the vector's **real** length
  - **setPoints** Set Vector's main points
  - **translate** Translate self vector to the given Point.
  - **unitVector** Get unit vector from self.
  - **vect** Get the absolute vector from origin

A full description of each Class is avaible on the left tree entries.

Here you will find the main Scheme to have a global look on this module.

CHAPTER

ONE

INTRODUCTION

1.1 Installation

To setup manually :

- **Install all prerequires as following :**
  - Get python3.8 from *www.python.org*
  - **pip install sphinx**
  - **pip install numpy**
  - **pip install cython**
  - **pip install sphinxcontrib-napoleon**
  - **pip install sphinx-autoapi**
- **Compile the algebra_3D.pyx file using the command : python setup_cython_3d.py build_ext –inplace**
- **compile the associated html file using the command: .make html**

CHAPTER

TWO

ALGEBRA 3D USAGE QUICK START

Here, you will find all the basics operations computed from the redefined python function.

2.1 Point Usage Quick Start

2.1.1 Add function

Ad hoc polymorphism ‘add’ function

Parameters	Type	Description
p	Point	The second operand as Point

Returns

Point The translated point.

Examples

```
>>> a=Point(1,1,1)
>>> b=Point(2,2,2)
>>> print(a+b)
 ( 3.0 , 3.0 , 3.0 )
printed
```

2.1.2 Sub function

Ad hoc polymorphism ‘subb’ function

Parameters	Type	Description
p	Point	The second operand as Point

Returns

Point The translated point.

Examples

```
>>> a=Point(1,1,1)
>>> b=Point(2,2,2)
>>> print(a-b)
 ( -1.0 , -1.0 , -1.0 )
printed
```

2.1.3 Neg function

Ad hoc polymorphism 'neg' function

Returns

Point The opposite point.

Examples

```
>>> p=Point(1,2,3)
>>> print(-p)
 ( -1.0 , -2.0 , -3.0 )
printed
```

2.1.4 Str function

Genericity polymorphism 'str function'

Examples

```
>>> p=Point(0,0,0)
>>> print(p)
 ( 0.0 , 0.0 , 0.0 )
printed
```

2.2 Vector Usage Quick Start

2.2.1 Add function

Ad hoc polymorphism 'add' function

Parameters	Type	description
vec	Vector	The second operand as vector

Returns

Vector The computed added vector

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> w=Vector(1,1,1,4,5,6)
>>> print(v+w)
 ( 5.0 , 6.0 , 7.0 )
printed
```

2.2.2 Sub function

Ad hoc polymorphism 'sub' function

Parameters	Type	description
vec	Vector	The second operand as vector

Returns

Vector The computed subbed vector

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> w=Vector(1,1,1,4,5,6)
>>> print(v-w)
 ( -1.0 , -2.0 , -3.0 )
printed
```

2.2.3 Mul function

Ad hoc polymorphism 'mul' function

Parameters	Type	description
vec	Vector	The second operand as vector

Returns

Vector The computed multiplied vector

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> print(v*10)
 ( 20.0 , 20.0 , 20.0 )
printed
```

2.2.4 Mod function

Ad hoc polymorphism 'mod' function

Parameters	Type	description
vec	Vector	The second operand as vector

Returns

double The length mod value

Examples

2.2.5 Neg function

Genericity polymorphism 'neg' function.

Returns

Vector The opposite vector

2.2.6 Contains function

Genericity polymorphism 'contains' function

Parameters	Type	Description
vec	Vector	The vector to compare

Returns

bint

Test the colinearity of the self vector and the second operand :

- True => self = alpha x vec
- False => self != alpha x vec

Examples

```
>>> v=Vector(0,0,0,1,1,1)
>>> w=Vector(0,0,0,2,2,2)
>>> z=Vector(1,1,1,2,3,4)
>>> print(v in w)
True
>>> print(v in z)
False
```

2.2.7 Str function

Genericity polymorphism 'str function'

2.2.8 Getitem function

Genericity polymorphism 'getitem' function

Parameters	Type	Description
key	int	the index of the value to get (must be < of the vecttor length)

Returns

double The indexed value

Examples

```
>>> v=Vector(0,0,0,1,2,3)
>>> print(v[0])
1.0
>>> print(v[1])
2.0
>>> print(v[2])
3.0
```

2.2.9 Setitem function

Genericity polymorphism 'setitem' function

Parameters	Type	Description
key	int	the index of the value to set (must be < of the vecttor length)
value	double	The value to set

Examples

```
>>> v=Vector(0,0,0,1,2,2)
>>> v[2]=3
>>> print(v)
 ( 1.0 , 2.0 , 3.0 )
printed
```

2.2.10 Eq function

Genericity polymorphism 'eq' function

Parameters	Type	Description
vec	Vector	The second vector to test

Returns

bint The equality test between the self vector object and the second vector as parameter

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> w=Vector(0,0,0,2,2,2)
>>> print(v==w)
True
```

2.2.11 Ne function

Genericity polymorphism 'ne' function

Parameters	Type	Description
vec	Vector	The second vector to test

Returns

bint The non-equality test between the self vector object and the second vector as parameter

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> z=Vector(0,0,0,1,2,3)
>>> print(v==z)
False
```

2.2.12 Le function

Genericity polymorphism 'le' function

Parameters	Type	Description
vec	Vector	The second vector to test

Returns

bint The Less or Equal test between the self vector object and the second vector as parameter

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> w=Vector(0,0,0,5,5,5)
>>> print(v<=w)
True
```

2.2.13 Lt function

Genericity polymorphism 'le' function

Parameters	Type	Description
vec	Vector	The second vector to test

Returns

bint The Less Than test between the self vector object and the second vector as parameter

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> w=Vector(0,0,0,5,5,5)
>>> print(v<w)
True
>>> print(w<v)
False
```

2.2.14 Ge function

Genericity polymorphism 'le' function

Parameters	Type	Description
vec	Vector	The second vector to test

Returns

bint The Greater or Equal test between the self vector object and the second vector as parameter

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> w=Vector(0,0,0,5,5,5)
>>> print(w>=v)
True
>>> print(v>=w)
False
```

2.2.15 Gt function

Genericity polymorphism 'le' function

Parameters	Type	Description
vec	Vector	The second vector to test

Returns

bint The Greater Than test between the self vector object and the second vector as parameter

Examples

```
>>> v=Vector(0,0,0,3,3,3)
>>> w=Vector(1,2,1,8,4,8)
>>> print(w>v)
True
>>> print(v>w)
False
```

CHAPTER

THREE

CLASS POINT

Here the main Point class engine definition.

algebra_3D.Point.copy	Copy an instance of the current Point object
algebra_3D.Point.op	Operator redefinition for the class Point.
algebra_3D.Point.print_screen	Utilitary screen printer : Print the Point coordonates.
algebra_3D.Point.set_x	Setters : Set the value of x
algebra_3D.Point.set_y	Setters : Set the value of y
algebra_3D.Point.set_z	Setters : Set the value of z
algebra_3D.Point.x	getters : Get the value of x
algebra_3D.Point.y	getters : Get the value of y
algebra_3D.Point.z	getters : Get the value of z

3.1 algebra_3D.Point.copy

Point.**copy**()

Copy an instance of the current Point object

Returns

Point The cloned Point object

See also:

Point.x

Point.y

Point.z

Examples

```
>>> p=Point(1,2,3)
>>> m=p.copy()
>>> print(p)
 ( 1.0 , 2.0 , 3.0 )
printed
>>> print(m)
 ( 1.0 , 2.0 , 3.0 )
printed
```

3.2 algebra_3D.Point.op

Point.**op**()

Operator redefinition for the class Point.

Parameters	Type	Description
B	Point	The second member as a Point
op	string	the operator to execute

Returns

Point The computed point operation

See also:

Point.x

Point.y

Point.z

Examples

```
>>> a=Point(1,1,1)
>>> b=a.copy()
>>> print(a)
 ( 1.0 , 1.0 , 1.0 )
printed
>>> print(b)
 ( 1.0 , 1.0 , 1.0 )
printed
>>> print(a.op(b,'+'))
 ( 2.0 , 2.0 , 2.0 )
printed
>>> print(a.op(b,'-'))
 ( 0.0 , 0.0 , 0.0 )
printed
```

3.3 algebra_3D.Point.print_screen

Point.**print_screen**()

Utilitary screen printer : Print the Point coordonates.

See also:

Point.x

Point.y

Point.z

Examples

```
>>> p=Point(1,2,3)
>>> p.print_screen()
 ( 1.0 , 2.0 , 3.0 )
```

3.4 algebra_3D.Point.set_x

Point.**set_x**()
Setters : Set the value of x

Parameters	Type	Description
x	double	The value to set

Examples

```
>>> p=Point(0,0,1)
>>> print(p)
 ( 0.0 , 0.0 , 1.0 )
printed
>>> p.set_x(2)
>>> print(p)
 ( 2.0 , 0.0 , 1.0 )
printed
```

3.5 algebra_3D.Point.set_y

Point.**set_y**()
Setters : Set the value of y

Parameters	Type	Description
y	double	The value to set

Examples

```
>>> p=Point(0,0,1)
>>> print(p)
 ( 0.0 , 0.0 , 1.0 )
printed
>>> p.set_y(2)
>>> print(p)
 ( 0.0 , 2.0 , 1.0 )
printed
```

3.6 algebra_3D.Point.set_z

Point.**set_z**()

Setters : Set the value of z

Parameters	Type	Description
z	double	The value to set

Examples

```
>>> p=Point(1,0,0)
>>> print(p)
 ( 1.0 , 0.0 , 0.0 )
printed
>>> p.set_z(2)
>>> print(p)
 ( 1.0 , 0.0 , 2.0 )
printed
```

3.7 algebra_3D.Point.x

Point.**x**()

getters : Get the value of x

Returns

double The value of x.

Examples

```
>>> p=Point(3,2,1)
>>> p.x()
3.0
```

3.8 algebra_3D.Point.y

Point.**y**()

getters : Get the value of y

Returns

double The value of y.

Examples

```
>>> p=Point(3,2,1)
>>> p.y()
2.0
```

3.9 algebra_3D.Point.z

Point.**z**()

getters : Get the value of z

Returns

double The value of z.

Examples

```
>>> p=Point(3,2,1)
>>> p.z()
1.0
```

CHAPTER

FOUR

CLASS VECTOR

Here the main Vector class engine definition.

algebra_3D.Vector.angle	Getters : Get the alpha angle between vector and norm.
algebra_3D.Vector.angle_vec	Get the angle between two vectors.
algebra_3D.Vector.copy	Copy an instance of the current Vector object
algebra_3D.Vector.cos	Get a fast computed cos from self vector and vec vector.
algebra_3D.Vector.dot	Dot product self and vec.
algebra_3D.Vector.hortogo	Test the hortogonal proprieties of self vector and given vec.
algebra_3D.Vector.length	Get the norm of self vector.
algebra_3D.Vector.op	Operator redefinition for the class Point.
algebra_3D.Vector.p1	Getters : Get the origin of vector as Point.
algebra_3D.Vector.p2	Getters : Get the destination of vector as Point.
algebra_3D.Vector.pointAt	Get the point result after a translation by self vector, scaling by t.
algebra_3D.Vector.print_screen	Utilitary screen printer.
algebra_3D.Vector.prod	Cross product self and vec.
algebra_3D.Vector.setLength	Change the length of the vector without lambda-multiplication (you can define a precise length).
algebra_3D.Vector.setP1	Setters : Set the origin of vector as Point
algebra_3D.Vector.setP2	Setters : Set the destination of vector as Point
algebra_3D.Vector.setPoints	Setters : Set the vector from Points arguments
algebra_3D.Vector.translate	Translate self vector to the given Point.
algebra_3D.Vector.unitVector	Get unit vector from self.
algebra_3D.Vector.vect	Get the absolute vector (from (0,0,0)).
algebra_3D.Vector.x1	Getters : Get the x coordonate of the origin vector.
algebra_3D.Vector.x2	Getters : Get the x coordonate of the destination vector.
algebra_3D.Vector.y1	Getters : Get the y coordonate of the origin vector.
algebra_3D.Vector.y2	Getters : Get the y coordonate of the destination vector.
algebra_3D.Vector.z1	Getters : Get the z coordonate of the origin vector.
algebra_3D.Vector.z2	Getters : Get the z coordonate of the destination vector.

4.1 algebra_3D.Vector.angle

Vector.**angle**()

Getters : Get the alpha angle between vector and norm. To fix : Please not use

Returns

double The alpha angle between vecttor and norm

4.2 algebra_3D.Vector.angle_vec

Vector.**angle_vec**()

Get the angle between two vectors. The operation is realized if and only if the two vectors have the same origin.

Parameters	Type	Description
vec	Vector	The vector angle-oriented with self vector
mode	int	**The angle unit :** • 0 => radians • 1 => degrees

Returns

double The angle between the two vectors.

See also:

Vector.cos

Examples

```
>>> u=Vector(0,0,0,1,0,0)
>>> v=Vector(0,0,0,0,1,0)
>>> u.angle_vec(v)
90.00003218077504
>>> u.angle_vec(v,0)
1.5707963267948966
```

4.3 algebra_3D.Vector.copy

Vector.**copy**()

Copy an instance of the current Vector object

Returns

Point The cloned Vector object

See also:

Vector.p1

Vector.p2

Point.x

Point.y

Point.z

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> w=v.copy()
>>> v==w
True
```

4.4 algebra_3D.Vector.cos

`Vector.cos()`

Get a fast computed cos from self vector and vec vector. The operation is realized if and only if the two vectors have the same origin.

Parameters	Type	Description
vec	Vector	The vector angle-oriented with self vector

Returns

double The computed cosine.

See also:

Vector.p1

Vector.vect

Vector.dot

Vector.length

Point.x

Point.y

Point.z

Examples

```
>>> v=Vector(0,0,0,1,0,0)
>>> u=Vector(0,0,0,0,1,0)
>>> v.cos(u)
0.0
```

4.5 algebra_3D.Vector.dot

Vector.**dot**()
Dot product self and vec.

Returns

double The dot result

See also:

Vector.vect

Vector.p1

Vector.p2

Point.x

Point.y

Point.z

Examples

```
>>> v=Vector(0,0,0,1,2,1)
>>> w=Vector(0,0,0,3,1,2)
>>> print(v.dot(w))
7.0
```

4.6 algebra_3D.Vector.hortogo

Vector.**hortogo**()
Test the hortogonal proprieties of self vector and given vec.

Parameters	Type	Description
vec	Vector	The vector to test with self

Returns

int

- 0 : Vectors are not hortgonal
- 1 : Vectors are hortogonal

See also:

Vector.dot

Examples

```
>>> u=Vector(0,0,0,1,0,0)
>>> v=Vector(0,0,0,0,1,0)
>>> print(u.hortogo(v))
True
```

4.7 algebra_3D.Vector.length

Vector.**length**()

Get the norm of self vector.

Returns

float The euclidian length of the vector

See also:

Vector.copy

Vector.vect

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> print(v.length())
3.4641016151377544
```

4.8 algebra_3D.Vector.op

Vector.**op**()

Operator redefinition for the class Point.

Parameters	Type	Description
B	Vector	The second member as a Vector
op	string	the operator to execute

Returns

Vector The computed vector operation

See also:

Vector.p1

Vector.p2

Point.x

Point.y

Point.z

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> w=Vector(1,1,1,4,4,4)
>>> print(v.op(w,'+'))
 ( 5.0 , 5.0 , 5.0 )
printed
>>> print(v.op(w,'-'))
 ( -1.0 , -1.0 , -1.0 )
printed
>>> p=Point(1,2,3)
>>> print(v.op(p,'*'))
 ( 2.0 , 4.0 , 6.0 )
printed
>>> print(v.op(10,'*'))
 ( 20.0 , 20.0 , 20.0 )
printed
```

4.9 algebra_3D.Vector.p1

Vector.**p1**()

Getters : Get the origin of vector as Point.

Returns

Point The origin of vector

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> v.p1()
<algebra_3D.Point object at 0x0B39F318>
>>> print(v.p1())
 ( 0.0 , 0.0 , 0.0 )
printed
```

4.10 algebra_3D.Vector.p2

Vector.**p2**()

Getters : Get the destination of vector as Point.

Returns

Point The destination of vector

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> print(v.p2())
 ( 2.0 , 2.0 , 2.0 )
printed
```

4.11 algebra_3D.Vector.pointAt

Vector.**pointAt**()

Get the point result after a translation by self vector, scaling by t.

Parameters	Type	Description
t	double	The scale range of the translation

Returns

Point The translated point.

See also:

Vector.p1

Vector.p2

Point.x

Point.y

Point.z

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> print(v.pointAt(1))
 ( 2.0 , 2.0 , 2.0 )
printed
>>> print(v.pointAt(2))
 ( 4.0 , 4.0 , 4.0 )
printed
>>> print(v.pointAt(3))
```

(continues on next page)

(continued from previous page)

```
 ( 6.0 , 6.0 , 6.0 )
printed
```

4.12 algebra_3D.Vector.print_screen

Vector.**print_screen**()
Utilitary screen printer.

4.13 algebra_3D.Vector.prod

Vector.**prod**()
Cross product self and vec.

Returns

double The prod result

See also:

Vector.copy

Vector.p1

Vector.p2

Point.x

Point.y

Point.z

Examples

```
>>> v=Vector(0,0,0,1,2,1)
>>> w=Vector(0,0,0,3,1,2)
>>> print(v.prod(w))
 ( 3.0 , 1.0 , -5.0 )
printed
```

4.14 algebra_3D.Vector.setLength

Vector.**setLength**()
Change the length of the vector without lambda-multiplication (you can define a precise length).

Parameters	Type	Description
length	double	The new length of the self vector
precision	int	The precision indice : bigger mean more precise

See also:

Vector.length

Vector.setP2

Vector.x2

Vector.y2

Vector.z2

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> print(v.length())
3.4641016151377544
>>> v.setLength(4)
>>> print(v)
 ( 2.3094020000000009 , 2.3094020000000009 , 2.3094020000000009 )
printed
>>> print(v.length())
4.0000015991011955
```

4.15 algebra_3D.Vector.setP1

`Vector.setP1()`
Setters : Set the origin of vector as Point

Parameters	Type	Description
x	Point	The origin of vector as Point

Examples

```
>>> v=Vector(0,0,0,1,2,3)
>>> p=Point(3,3,3)
>>> v.setP1(p)
>>> print(v)
 ( -2.0 , -1.0 , 0.0 )
printed
```

4.16 algebra_3D.Vector.setP2

`Vector.setP2()`
Setters : Set the destination of vector as Point

Parameters	Type	Description
y	Point	The origin of vector as Point

Examples

```
>>> v=Vector(0,0,0,1,2,3)
>>> p=Point(3,3,3)
>>> v.setP2(p)
>>> print(v)
 ( 3.0 , 3.0 , 3.0 )
printed
```

4.17 algebra_3D.Vector.setPoints

Vector.**setPoints**()
Setters : Set the vector from Points arguments

Parameters	Type	Description
x	Point	The origin of vecttor as Point
y	Point	The destination of vector as Point

See also:

Vector.setP1

Vector.setP2

Examples

```
>>> v=Vector(0,0,0,1,2,3)
>>> p=Point(0,0,0)
>>> m=Point(3,2,1)
>>> v.setPoints(p,m)
>>> print(v)
 ( 3.0 , 2.0 , 1.0 )
printed
```

4.18 algebra_3D.Vector.translate

Vector.**translate**()
Translate self vector to the given Point.

Parameters	Type	Description
point	Point	The translation destination point

See also:

Vector.vect

Vector.x2

Vector.y2

Vector.z2

Point.x

Point.y

Point.z

Examples

```
>>> v=Vector(0,0,0,2,2,2)
>>> print(v.translate(Point(1,1,1)).p1())
 ( 1.0 , 1.0 , 1.0 )
printed
>>> print(v.translate(Point(1,1,1)).p2())
 ( 3.0 , 3.0 , 3.0 )
printed
```

4.19 algebra_3D.Vector.unitVector

Vector.**unitVector**()

Get unit vector from self.

Returns

Double The unit vector from self (divided by the enclidean length)

See also:

Vecotr.copy

Vector.length

4.20 algebra_3D.Vector.vect

Vector.**vect**()

Get the absolute vector (from (0,0,0)).

Returns

Vector The absolute vector from the self object

See also:

Vector.x1

Vector.x2

Vector.y1

Vector.y2

Vector.z1

Vector.z2

Examples

```
>>> v=Vector(1,1,1,3,3,3)
>>> print(v.vect())
 ( 2.0 , 2.0 , 2.0 )
printed
```

4.21 algebra_3D.Vector.x1

Vector.**x1**()

Getters : Get the x coordonate of the origin vector.

See also:

Point.x

Examples

```
>>> v=Vector(1,2,3,4,5,6)
>>> v.x1()
1.0
```

4.22 algebra_3D.Vector.x2

Vector.**x2**()

Getters : Get the x coordonate of the destination vector.

See also:

Point.x

Examples

```
>>> v=Vector(1,2,3,4,5,6)
>>> v.x2()
4.0
```

4.23 algebra_3D.Vector.y1

Vector.**y1**()

Getters : Get the y coordonate of the origin vector.

See also:

Point.y

Examples

```
>>> v=Vector(1,2,3,4,5,6)
>>> v.y1()
2.0
```

4.24 algebra_3D.Vector.y2

Vector.**y2**()
Getters : Get the y coordonate of the destination vector.

See also:

Point.y

Examples

```
>>> v=Vector(1,2,3,4,5,6)
>>> v.y2()
5.0
```

4.25 algebra_3D.Vector.z1

Vector.**z1**()
Getters : Get the z coordonate of the origin vector.

See also:

Point.z

Examples

```
>>> v=Vector(1,2,3,4,5,6)
>>> v.z1()
3.0
```

4.26 algebra_3D.Vector.z2

Vector.**z2**()
Getters : Get the z coordonate of the destination vector.

See also:

Point.z

Examples

```
>>> v=Vector(1,2,3,4,5,6)
>>> v.z2()
6.0
```

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex
- search

INDEX