05 libaldo math intro

July 28, 2021

[1]:
```
from sympy import *
from polyclass import *
from libaldo_math import *
from libaldo_show import *
from physic_lib import *
from IPython.display import display, Math
init_printing()
```

0.0.1 Introduccion to libaldo_math library

The libaldo_math library includes several functions that will make the handling of equations and their solution easier. Although there are several ways to apply regular functions I have decided to include a short explanation as they make my life easier when solving algebraic problems, it is worth mentioning that all these functions are based on the sympy library which until today I think is the best of it. better but I think or have not yet learned how to handle variables and try to express them without using the Latex language which for me is very complicated, I hope it will be helpful

In this part I am going to separate it into 4 subparts and they are:

```
* regulars functions
* opemat()  used to simplify
* csolve() used to solve equations
* fpoly() to get a lot of equations propieties and change
```

Of course i will try to compare whit sympy method..

Regular functions()

[2]:
```
# frs():return numeric but symbolyc fraction
# regular form
3/5
```

[2]: 0.6

[3]:
```
frs(3,5) # with libaldo_mat
```

[3]: $\frac{3}{5}$

[4]:
```
1+2/3
```

[4]: 1.66666666666667

[5]:
```
1+frs(2,3)
```

[5]: $\frac{5}{3}$

[6]:
```
# pow number
2**(2/7)
```

[6]: 1.21901365420448

[7]:
```
pow(2,2/7)
```

[7]: 1.21901365420448

[10]:
```
kpow(2,frs(2,7))# with libaldo_mat
```

[10]: $2^{\frac{2}{7}}$

[12]:
```
sqrt(2/32)
```

[12]: 0.25

[13]:
```
rpow(2,32) # with libaldo_mat
```

[13]: $\sqrt[32]{2}$

[14]:
```
sex2rad(90) # sexagesimal to radians
```

[14]: $\frac{\pi}{2}$

[15]:
```
rad2sex(pi/2)
```

[15]: 90.0

0.0.2 opemat() function is used to simplyfy follo the secuense string...

[16]:
```
x,y,alpha=symbols('x y alpha')
```

[28]:
```
Eq1= kpow(3*x+y,2)+kpow(-x*y+6*y+5*x,2)
Eq1
```

[28]: $(3x + y)^2 + (-xy + 5x + 6y)^2$

[32]:
```
Eq1= kpow(3*x+y,2)+kpow(-x-2*y,2)
Eq1
```

[32]: $(-x - 2y)^2 + (3x + y)^2$

[34]:
```
opemat(Eq1,'ef') # e=expand f=factor
```

[34]:
$$5\left(2x^2+2xy+y^2\right)$$

[36]:
```
opemat(Eq1,'es') # e=expand s=simplify
```

[36]:
$$10x^2+10xy+5y^2$$

[37]:
```
Eq2=2*sin(alpha)+3*cos(2*alpha)
Eq2
```

[37]:
$$2\sin\left(\alpha\right)+3\cos\left(2\alpha\right)$$

[38]:
```
opemat(Eq2,'xt') # trig expand and later
```

[38]:
$$2\sin\left(\alpha\right)+6\cos^2\left(\alpha\right)-3$$

[28]:
```
Eq1= kpow(3*x+y,2)+kpow(-x*y+6*y+5*x,2)
Eq1
```

[28]:
$$(3x+y)^2+(-xy+5x+6y)^2$$

0.0.3 part(), inpart(), cpart()

[39]:
```
# I hope the examples explain themselves that they do these functions
# creating simple Equation
Eq1= kpow(x,y)*rpow(3*y,2*x)/(kpow(sin(alpha),3))
Eq1
```

[39]:
$$\frac{x^y\,(3y)^{\frac{x}{2}}}{\sin^3\left(\alpha\right)}$$

[40]:
```
# part(expr,address)
part(Eq1,[0])
```

[40]:
$$x^y$$

[43]:
```
part(Eq1,[0,0])
```

[43]:
$$x$$

[44]:
```
part(Eq1,[0,1])
```

[44]:
$$y$$

[45]:
```
part(Eq1,[1])
```

[45]:
$$(3y)^{\frac{x}{2}}$$

[46]:
```
part(Eq1,[1,0])
```

[46]:
$$3y$$

[47]:
```
part(Eq1,[1,1])
```

[47]:
$$\frac{x}{2}$$

[48]:
```
part(Eq1,[2])
```

[48]:
$$\frac{1}{\sin^3(\alpha)}$$

[51]:
```
part(Eq1,[2,0,0])
```

[51]:
$$\alpha$$

[54]:
```
# inpart(expr,repl,address)
beta=symbols('beta')
Eq1
```

[54]:
$$\frac{x^y (3y)^{\frac{x}{2}}}{\sin^3(\alpha)}$$

[56]:
```
inpart(Eq1,beta,[2,0,0])
```

[56]:
$$\frac{x^y (3y)^{\frac{x}{2}}}{\sin^3(\beta)}$$

[57]:
```
# cpart(expr,address) Show where are the expresion choice
cpart(Eq1,[2,0,0])
```

[57]:
$$Set\left(\frac{x^y (3y)^{\frac{x}{2}}}{\sin^3(PART)}, \alpha\right)$$

[58]:
```
cpart(Eq1,[1,1])
```

[58]:
$$Set\left(\frac{x^y (3y)^{PART}}{\sin^3(\alpha)}, \frac{x}{2}\right)$$

0.0.4 csolve() is used to solve equations

csolve() is based in solve and solveset from sympy but whit a little change to get most usefull answer and if you want nice answer.. for exaple

[61]:
```
Eq3= 3*x+15*y-20; Eq3 # we know that in python and sympy single eq if like␣
↪equal cero
```

[61]:
$$3x + 15y - 20$$

[64]:
```
# with sympy the return is a list and maybe notr useful in this moment
solve(Eq3,x)
```

[64]:
$$\left[\frac{20}{3} - 5y\right]$$

[65]:
```
# with libaldo math
csolve(Eq3,x)
```

[65]:
$$\frac{20}{3} - 5y$$

[66]:
```
# also we can store in new val and retun nice answer like
x=csolve(Eq3,x,'x')
```

$$x = \frac{20}{3} - 5y$$

[68]:
```
x,y,alpha=symbols('x y alpha')
```

[69]:
```
Eq2=2*sin(alpha)+3*cos(2*alpha)-5*x
Eq2
```

[69]:
$$-5x + 2\sin(\alpha) + 3\cos(2\alpha)$$

[70]:
```
csolve(Eq2,x)
```

[70]:
$$\frac{2\sin(\alpha)}{5} + \frac{3\cos(2\alpha)}{5}$$

[74]:
```
x=csolve(Eq2,x,'x',kope='x') # solve x ,nice 'x=', simplyfy expand and sim trig
```

$$x = \frac{2\sin(\alpha)}{5} + \frac{6\cos^2(\alpha)}{5} - \frac{3}{5}$$

fpoly() is used to get info and maybe manage Eq and some internal function descrbe here

[ ]:
```
x
```

[ ]:
$$\frac{2\sin(\alpha)}{5} + \frac{6\cos^2(\alpha)}{5} - \frac{3}{5}$$

[ ]:
```
fpoly(x,'list') # list args Eq
```

[ ]:
$$\left[-\frac{3}{5}, \frac{2\sin(\alpha)}{5}, \frac{6\cos^2(\alpha)}{5}\right]$$

[ ]:
```
fpoly(x,'n') # num term Eq
```

[ ]:
$$3$$

[ ]:
```
fpoly(x,'free') # list symbols in  Eq
```

[ ]:
$[\alpha]$

Here some fpoly optionssss

[2]:
```
x,y,alpha=symbols('x y alpha') # some sampless
```

[3]:
```
Eq4=2*sin(alpha)+3*cos(2*alpha)-5*x+30*y
Eq4
```

[3]:
$-5x + 30y + 2\sin(\alpha) + 3\cos(2\alpha)$

[4]:
```
fpoly(Eq4,'free') # list of symbols  in  Eq4
```

[4]:
$[y,\ x,\ \alpha]$

[5]:
```
fpoly(Eq4,'n') # numargs like len(Eq4.args) whit sympy
```

[5]:
4

[6]:
```
fpoly(Eq4,'list') # list of args in Eq4
```

[6]:
$[-5x,\ 2\sin(\alpha),\ 3\cos(2\alpha),\ 30y]$

[7]:
```
fpoly(Eq4,'get',1) # get arg No 1
```

[7]:
$2\sin(\alpha)$

[8]:
```
x
```

[8]:
x

[9]:
```
fpoly(x,'list') # list args Eq
```

[9]:
$[x]$

[10]:
```
fpoly(x,'n') # num term Eq
```

[10]:
1

[11]:
```
fpoly(x,'free') # list symbols in  Eq
```

[11]:
$[x]$

[12]:
```
x,y,alpha=symbols('x y alpha')
```

[13]:
```
Eq4=2*sin(alpha)+3*cos(2*alpha)-5*x+30*y
Eq4
```

[13]:
$-5x + 30y + 2\sin(\alpha) + 3\cos(2\alpha)$

[14]:
```
fpoly(Eq4,'free') # list symbols in  Eq
```

[14]:
$$[y,\ x,\ \alpha]$$

[15]:
```
fpoly(Eq4,'n') # numargs like len(Eq4.args) whit sympy
```

[15]:
$$4$$

[16]:
```
fpoly(Eq4,'list') # list of args in Eq4
```

[16]:
$$[-5x,\ 2\sin(\alpha),\ 3\cos(2\alpha),\ 30y]$$

[17]:
```
fpoly(Eq4,'get',1) # get arg No 1
```

[17]:
$$2\sin(\alpha)$$

[ ]:
```

```