simone_1

December 10, 2020

1 Introduction to python and sympy

In this lecture we'll see the basics of python programming, focusing on the sympy module. Sympy is a CAS (Computer Algebra System), with many useful functionalities:

- Solution of algebraic equations
- Limits, derivatives, integrals
- Linear algebra
- 2D and 3D plot
- and many more...

for further reference see the sympy website

1.1 Introduction to the python programming language

Here we familiarize with the python language. We'll see how to:

- Define variables
- Combine them
- Understand and use data types
- Print variables on the screen
- Control the program flow with if, for, while statements
- Define functions

1.1.1 Basics

[1]:
```
a = 3 # define a number
a1, a2 = 10, 45
s1 = "Physics" # define a string
s2 = "Mathematics" # anothe string
s3 = "Architecture" # your favourite string!
s4 = "Python" # today's string
b = (s1 == s2) # define a boolean (True or False)
```

[2]:
```
# We can print our variables
print("Hello!") # printing a string
print("Good morning students!","This is a lecture on", s4) # printing more␣
 ↪arguments
```

```
Hello!
Good morning students! This is a lecture on Python
```

[3]:
```
print("We can print other types of variables as well:")
print(a, "is the magic number")
print("Is",s1,"the same thing as",s2, "?", b)
```

```
We can print other types of variables as well:
3 is the magic number
Is Physics the same thing as Mathematics ? False
```

[4]:
```
print("Now let's combine our variables!")
x1= 0.4
x2 = 7.9
print(x1, "-", x2, "=", x1-x2)

p = "Congratulations! You passed the exam with " + "30/30"
print(p)
nA = len(s3) # length of the string (number of characters)
print("The string '",s3,"' contains", nA, "characters")
```

```
Now let's combine our variables!
0.4 - 7.9 = -7.5
Congratulations! You passed the exam with 30/30
The string ' Architecture ' contains 12 characters
```

[5]:
```
print("We can find out the data type of our variable:")
print("The type of 'a' is;", type(a))
x = 29.8
print("The type of 'x' is", type(x))
print("'b' is a", type(b))
```

```
We can find out the data type of our variable:
The type of 'a' is; <class 'int'>
The type of 'x' is <class 'float'>
'b' is a <class 'bool'>
```

[6]:
```
print("We can automate our processes defining functions")
def introduce(name):
  print("Hi, my name is", name)

print("We can call the function as many times we want")
introduce("Laura")
introduce("Camillo")
introduce("Simone")
```

```
We can automate our processes defining functions
We can call the function as many times we want
Hi, my name is Laura
Hi, my name is Camillo
Hi, my name is Simone
```

1.1.2 Control flow

[7]:
```
print("Boolean values can control the conditional execution of some piece of␣
 ↪code:")
if 3>0:
  print("3 is greater than 0")
else:
  print("3 is less than 0")
```

```
Boolean values can control the conditional execution of some piece of code:
3 is greater than 0
```

[8]:
```
print("Now let's try with something non-trivial")
from datetime import date # module for date and time
today = date.today()
print("Today's date:", today)

print("Note:", type(today), "is not a string")
AMD = str(today).split("-") # converting to string and splitting
D =  int(AMD[2]) # converting to int
if D%2==0: # D is even
  print("Today is an even day")
else:
  print("Today is an odd day")
```

```
Now let's try with something non-trivial
Today's date: 2020-12-10
Note: <class 'datetime.date'> is not a string
Today is an even day
```

[9]:
```
print("Now let's check greek pi")
from numpy import pi as pi
if type(pi)==int:
  print(pi,"is an integer")
else:
  print(pi, "is not an integer")
```

```
Now let's check greek pi
3.141592653589793 is not an integer
```

[10]:
```
print("'for' loop")
for f in ["Al","John","Jack"]:    # ciclo for
```

```
    print("Good morning,", f, ", how are you?")
```

```
'for' loop
Good morning, Al , how are you?
Good morning, John , how are you?
Good morning, Jack , how are you?
```

[11]:
```
print("'while' loop")
T=0
print("You can start your test")
while T<7:
  print(7-T, "minutes left")
  T = T + 1
# end of while
print("Time is up!")
```

```
'while' loop
You can start your test
7 minutes left
6 minutes left
5 minutes left
4 minutes left
3 minutes left
2 minutes left
1 minutes left
Time is up!
```

1.2 sympy library

Now we explore the functionalities of sympy

[12]:
```
from sympy import * # we import the entire sympy
from sympy.plotting import *
```

[13]:
```
x, y, z = symbols('x y z') # defining symbols
```

1.2.1 solutiuon of algebraic equations

[14]:
```
print("We can solve algebraic equations")
S1 = solve(x**2-4, x)
print(S1)
for s in S1:
  display(Eq(x,s))
```

```
We can solve algebraic equations
[-2, 2]
```

$x = -2$

$x = 2$

[15]:
```
eqn = Eq(x**2-y+1, 0)
print("We want to solve the following equation:")
display(eqn)
print("The solutions are:")
S2 = solve(eqn,x)
for s in S2:
  display(Eq(x,s))
```

```
We want to solve the following equation:
```

$x^2 - y + 1 = 0$

```
The solutions are:
```

$x = -\sqrt{y - 1}$

$x = \sqrt{y - 1}$

1.2.2 Plotting

Sympy can be used for plotting

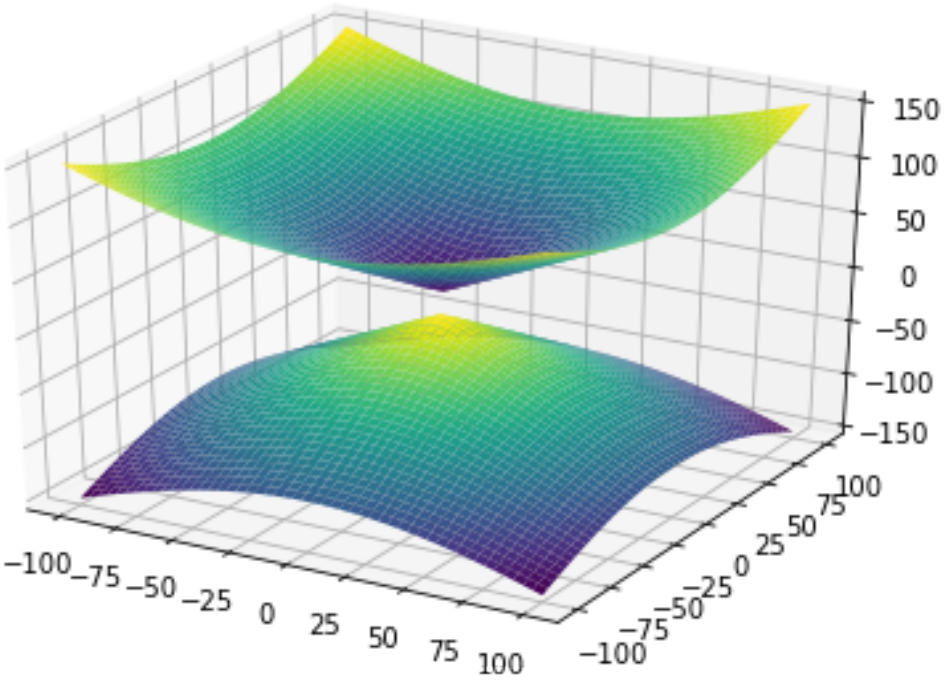
[16]:
```
print("Paraboloids:")
z0 = 10
PRB = z0 + sqrt(x**2+y**2)
z1 = PRB
z2 = -PRB
display(Eq(z, z1),Eq(z, z2))

L = 100
plot3d(z1,z2, (x,-L,L),(y,-L,L))
```

```
Paraboloids:
```

$z = \sqrt{x^2 + y^2} + 10$

$z = -\sqrt{x^2 + y^2} - 10$



[16]: <sympy.plotting.plot.Plot at 0x7fd9a5582910>

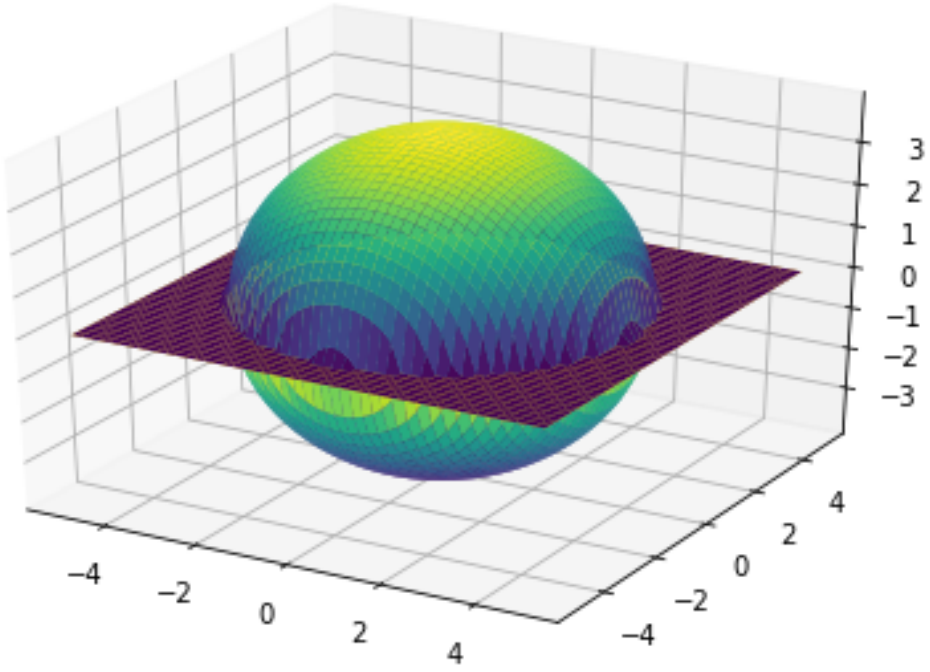
[17]:
```
def plot3d_quadric(eq,x,a,b,y,c,d,z):
    sol = solve(eq,z) # solving for 'z'
    print("Plotting the following curves:")
    for s in sol:
      display(Eq(z,s))
    plot3d(sol[0],sol[1],(x,a,b),(y,c,d))
```

[18]:
```
plot3d_quadric(x**2+y**2+z**2-16,x=x,a=-5,b=5,y=y,c=-5,d=5,z=z)
```

Plotting the following curves:

$$z = -\sqrt{-x^2 - y^2 + 16}$$

$$z = \sqrt{-x^2 - y^2 + 16}$$



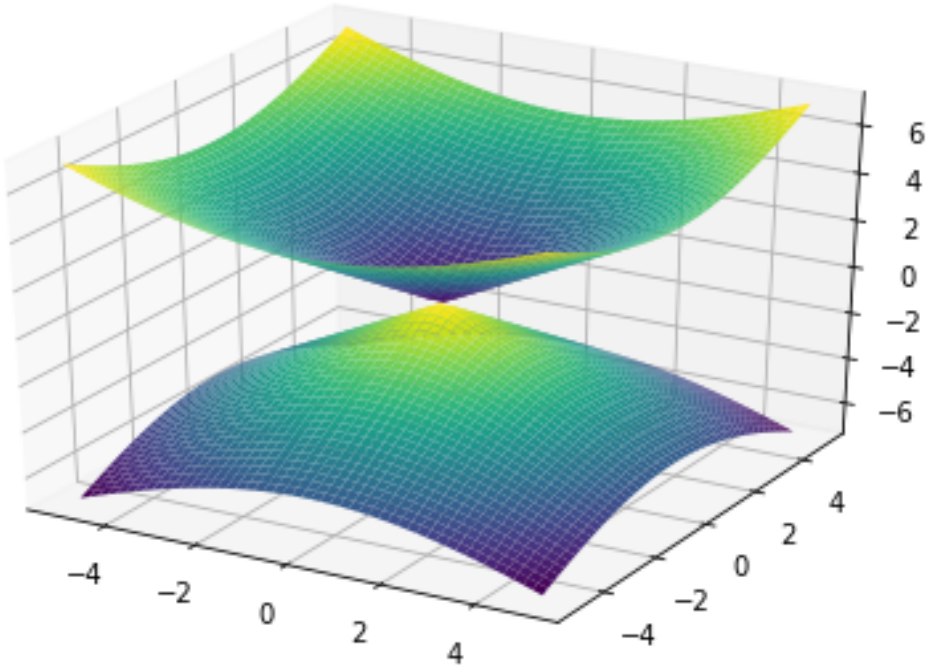
[19]:
```
Q1 = z**2-x**2-y**2
display(Eq(Q1, 0))
plot3d_quadric(z**2-x**2-y**2,x,-5,5,y,-5,5,z)
```

$-x^2 - y^2 + z^2 = 0$

Plotting the following curves:

$z = -\sqrt{x^2 + y^2}$

$z = \sqrt{x^2 + y^2}$



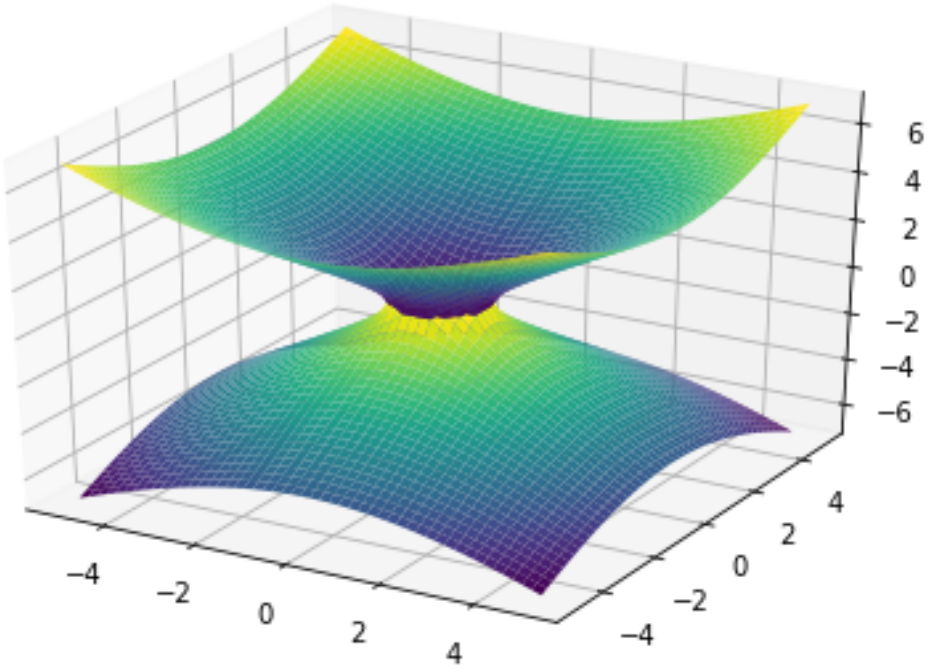
[20]:
```
Q2 = x**2+y**2-z**2-1
display(Eq(Q2, 0))
print("Hyperboloid from the 2 z(x,y) curves:")
plot3d_quadric(Q2,x,-5,5,y,-5,5,z)
```

$$x^2 + y^2 - z^2 - 1 = 0$$

```
Hyperboloid from the 2 z(x,y) curves:
Plotting the following curves:
```

$$z = -\sqrt{x^2 + y^2 - 1}$$

$$z = \sqrt{x^2 + y^2 - 1}$$



Parametric plots We now consider the hyperboloid as a ruled surface. The equation is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1$$

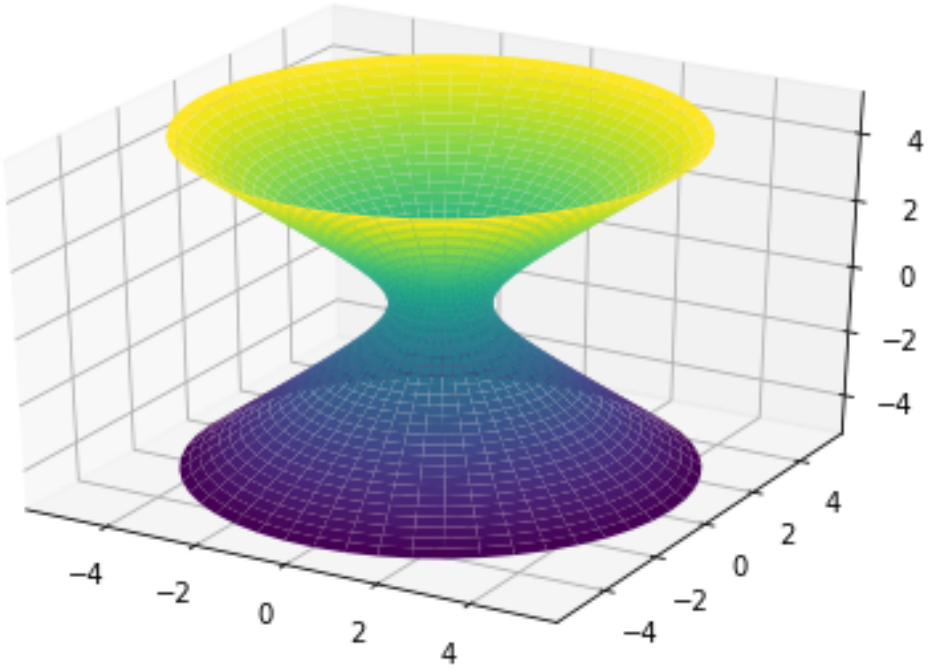
the parametrized version is:

$$x(t,s) = a\sqrt{1+t^2}\,\cos s$$
$$y(t,s) = b\sqrt{1+t^2}\,\sin s$$
$$z(t,s) = ct$$

for simplicity we use $a = b = c = 1$.

[21]:
```
t,s = symbols('t, s')
a=b=c=1
from numpy import pi as pi
n=20
print("That's your hyperbolid!")
plot3d_parametric_surface(a*sqrt(1+t**2)*cos(s),b*sqrt(1+t**2)*sin(s), c*t, (t,␣
↪-5, 5), (s,0,2*pi))
```

```
That's your hyperbolid!
```



[21]: <sympy.plotting.plot.Plot at 0x7fd99df7e6d0>

[ ]: