Computer Science Project

VSEPR in Python

Hardik Mirg XI-E

Index

Introduction

Valence Shell Electron Pair Repulsion (VSEPR) Theory of Inorganic Chemistry is a model for predicting the 3D shape of molecules based on the number of valence electron pairs around the central atom. It is a simple but powerful tool that can be used to explain a wide variety of molecular structures, from simple diatomic molecules like hydrogen chloride (HCl) to complex molecules like caffeine.

In this project, we will use Python to implement a **VSEPR model** that can **predict the 3D shape of molecules** based on their Lewis structure or Molecular Formula. Our model will be able to handle a wide variety of molecular structures, including molecules with **lone pairs, and multiple bonds**.

Our VSEPR Python Program is a **Menu-Driven Program** awaiting input from the user.

It has Four Modes including:

- VSEPR Calculator
- VSEPR Quiz
- Hybridized Skeletal Structures
- The Periodic Table

The Program uses **Special Characters to Draw Molecular Structures** and provide a better way to understand them.

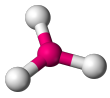
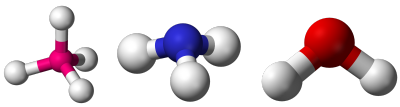
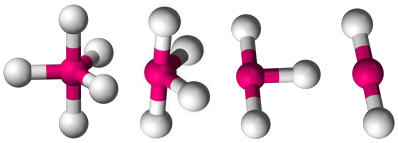
Most of the features offered by python were used while making this project. Some of them include **Strings, Lists, Dictionaries, Modules such as Math and Random, Loops, Functions, Conditional Statements, etc.**

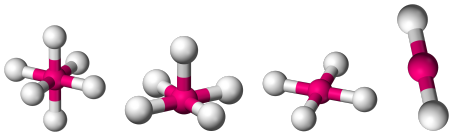
Prerequisites

It is recommended to have basic knowledge of the following topics:

- Periodic Table
- Valence Electrons and Lone Pairs
- Various Hybridizations, Geometries, and Shapes

Information Required to use the Program:

Hybridization	Geometry	Lone Pairs	Structure
sp	Linear	0 Shape:	 Regular
sp2	Trigonal Planar	0 Shape:	 Regular
sp3	Tetrahedral	0,1,2 Shape:	 Regular Pyramidal Bent
sp3d	Trigonal Bipyramidal	0,1,2,3 Shape:	 Regular See Saw T Shape Linear

Hybridization	Geometry	Lone Pairs	Structure
sp3d2	Octahedral	0,1,2,4 Shape:	 Regular Square Pyramidal Square Planar Linear

Operating Instructions

- Install the latest version of **Python** from here and make sure you have the necessary libraries installed
- **Download and Run** the Python **Code** from this github repository
- Take care of **Case Sensitivity** while entering **symbols of elements** so as to avoid ambiguity
  E.g: To enter the symbol for Cobalt: Co ✅ CO ❌
- Enter **Molecular Formulae** in the given format:
  E.g: To enter formula for Carbon Dioxide:
  CO2 ❌C_O*2 ✅
  Every atom should be separated with an underscore symbol (_) while an asterisk (*) symbol should be used to show how many times an atom is repeated
- **Charge** on a compound should be entered as an Integer
  E.g: 2- ❌ -2 ✅
- Make sure that there are **no typing errors** especially in the Quiz Mode as it would be considered to be the **wrong answer**
  E.g: Linear ✅Liner ❌
  Only the words provided in the earlier table are considered correct by the program

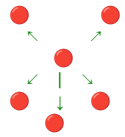
Code

```
```

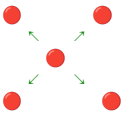
```
pt = '''
   H                                                  He
   Li Be                                 B    C    N  O   F   Ne
   Na Mg                                 Al   Si   P  S   Cl  Ar
   K  Ca Sc Ti  V  Cr Mn  Fe  Co  Ni  Cu Zn  Ga   Ge  As  Se  Br  Kr
   Rb Sr  Y Zr  Nb Mo Tc  Ru  Rh  Pd  Ag Cd  In   Sn  Sb  Te  I   Xe
   Cs Ba  | Hf  Ta W  Re  Os  Ir  Pt  Au Hg  Tl   Pb  Bi  Po  At  Rn
   Fr Ra  | Rf  Db Sg Bh  Hs  Mt  Ds  Rg Cn  Uut Uuq Uup Uuh Uus Uuo
          |
          ↳ La Ce  Pr Nd Pm  Sm Eu Gd Tb Dy  Ho  Er  Tm  Yb  Lu
          ↳ Ac Th  Pa U  Np  Pu Am Cm Bk Cf  Es  Fm  Md  No  Lr'''


print("\nVALANCE SHELL ELECTRON PAIR REPULSION THEORY ~ VSEPR")
print("~By Hardik Mirg")

PT = [
[["H"],[""],[""],[""],[""],[""],[""],[""],[""],[""],[""],[""],[""],[""],[""],[""],[""],["He"]],
[["Li"],["Be"],[""],[""],[""],[""],[""],[""],[""],[""],[""],[""],["B"],["C"],["N"],["O"],["F"],["Ne"]],
[["Na"],["Mg"],[""],[""],[""],[""],[""],[""],[""],[""],[""],[""],["Al"],["Si"],["P"],["S"],["Cl"],["Ar"]],
[["K"],["Ca"],["Sc"],["Ti"],["V"],["Cr"],["Mn"],["Fe"],["Co"],["Ni"],["Cu"],["Zn"],["Ga"],["Ge"],["As"],["Se"],["Br"],["Kr"]],
[["Rb"],["Sr"],["Y"],["Zr"],["Nb"],["Mo"],["Tc"],["Ru"],["Rh"],["Pd"],["Ag"],["Cd"],["In"],["Sn"],["Sb"],["Te"],["I"],["Xe"]],
[["Cs"],["Ba"],[""],["Hf"],["Ta"],["W"],["Re"],["Os"],["Ir"],["Pt"],["Au"],["Hg"],["Tl"],["Pb"],["Bi"],["Po"],["At"],["Rn"]],
[["Fr"],["Ra"],[""],["Rf"],["Db"],["Sb"],["Bh"],["Hs"],["Mt"],["Ds"],["Rg"],["Cn"],["Uut"],["Uuq"],["Uup"],["Uuh"],["Uus"],["Uuo"]]]

# To find group and period of element

def location(e):

    for j in range(7):
        for i in range(18):
            if PT[j][i][0] == e:
                return([i+1, j+1]) # returns group, period

# Returns the no. of valence electrons of an element
def ve (e):
    pos = location(e)
    if pos[0] <= 2: # if group <= 2 (s block)
        return pos[0]
    elif pos[0] >= 13: # if group >= 13 (p block)
        return (pos[0] - 10)

def valency(e):
    val_e = ve(e)

    if val_e < 4:
        return val_e
    elif val_e > 4:
        return (8-val_e)
    else:
        return 4

skeletal_str = {
"sp": ['''🔴←🔴→🔴'''],
"sp2":['''
       🔴
       ↑
       🔴
     ↙   ↘
   🔴       🔴
''',
```

```
'''
    ●
  ↙   ↘
●       ●
'''],
"sp3":['''
        ●
        ↑
        ●
      ↙ ↓ ↘
    ●   ●   ●

''',
'''
        ●
      ↙ ↓ ↘
    ●   ●   ●

''',
'''
        ●
    ↙       ↘
●               ●

'''],
"sp3d":['''
        ●
    ●   ↑
      ↖ |
        ● → ●
    ↙   |
    ●   ↓
        ●

''',
'''
        ●
    ●   ↑
      ↖ |
        ●
    ↙   |
    ●   ↓
        ●

''',

'''
    ●
    ↑
    |
  ● → ●
    |
    ↓
    ●

''',
'''
    ●
    ↑
    |
    ●
    |
    ↓
    ●

'''],
"sp3d2":['''
        ●
    ●   ↑   ●
      ↖ |   ↗
        ●
    ↙   |   ↘
```

```
''',
'''
```



```
''',
'''
```



```
'''],
"sp3d3":["Too complex to represent this 3D structure on a 2D screen"]
}

def structure (hyb, lp):
    return skeletal_str[hyb][lp]
mode = ""
score = 0
def welcome():
    print("\nSelect Application Mode:")
    print('''
    Enter (1) for VSEPR Calculator
    Enter (2) for VSEPR Quiz
    Enter (3) for Hybridized Skeletal Structures
    Enter (4) for Periodic Table

    ''')
    mode = int(input("Enter mode:\t"))
    if mode == 1:
        calculator()
    elif mode == 2:
        quiz()
    elif mode == 3:
        choice = "y"
        valid_hyb = ["sp", "sp2", "sp3", "sp3d", "sp3d2", "sp3d3"]
        while choice == "y":
            hyb = input("Enter Hybridization of the Structure: ").lower()
            lp = int(input("Enter number of Lone Pairs"))
            if valid_hyb.count(hyb) == 0:
                print("Please Enter Valid Hybridization")
            else:
                print("Skeletal Structure:", structure(hyb, lp), sep="\n")
            choice = input("Want To Start Again? (Y or N): ").lower()
        welcome()
    elif mode == 4:
        print(pt)
        a = input()
        if a == "" or a != "":
            welcome()
    else:
        print('Please Enter a Valid Mode Number:(')

final_sol = []
def mech(c, q):


    f = c.split('_')
    e_ = 0
    s_atm = 0
    s_atm_l = []
    c_atm = ""

    bond_e = 0
```

```
if len(f) == 1:
    if f[0].find("*") != -1:
        j = f[0].split("*")
        valance_e = ve(j[0]) * int(j[1])
        e_ += valance_e
        bond_e += valency(j[0]) * (int(j[1])-1)
        c_atm = j[0]
        s_atm += (int(j[1]) -1)
else:

    for i in f:
        if i.find("*") != -1:
            j = i.split("*")
            valance_e = ve(j[0]) * int(j[1])
            e_ += valance_e
            bond_e += valency(j[0]) * int(j[1])
            s_atm += int(j[1])
            s_atm_l.append(j[0])
        else:
            valance_e = ve(i)
            e_ += int(valance_e)
            if i == "H":
                bond_e += valency("H")
                s_atm+=1
            else:
                c_atm = i

def LP (cen_atm):
    return (((ve(cen_atm) - q) - bond_e)/2)

lp_c_atm = LP(c_atm)
s_no = s_atm + lp_c_atm


h_map = {
    2: ["sp", "Linear"],
    3: ["sp2", "Trigonal Planar"],
    4: ["sp3", "Tetrahedral"],
    5: ["sp3d", "Trigonal Bipyramidal"],
    6: ["sp3d2", "Octahedral"],
    7: ["sp3d3", "Pentagonal Bipyramidal"]}

def shape (hyb, lp):
    if hyb == "sp":
        return "Regular"
    elif hyb == "sp2":
        if lp == 1:
            return "Bent"
        else:
            return "Regular"
    elif hyb == "sp3":
        if lp == 1:
            return "Pyramidal"
        elif lp == 2:
            return "Bent"
        else:
            return "Regular"
    elif hyb == "sp3d":
        if lp == 1:
            return "See Saw"
        elif lp == 2:
            return "T Shape"
        elif lp == 3:
            return "Linear"
        else:
            return "Regular"
    elif hyb == "sp3d2":
        if lp == 1:
            return "Square Pyramidal"
```

```
            elif lp == 2:
                return "Square Planar"
            elif lp == 4:
                return "Linear"
            else:
                return "Regular"
        elif hyb == "sp3d3":
            if lp == 1:
                return "Pentagonal Pyramidal"
            elif lp == 2:
                return "Pentagonal Planar"
            elif lp == 5:
                return "Linear"
            else:
                return "Regular"


    fin = [h_map[s_no][0], h_map[s_no][1], shape(h_map[s_no][0], lp_c_atm), lp_c_atm]
    final_sol.clear()
    for i in fin:
        final_sol.append(i)
    return final_sol


def calculator ():

    print("\n")
    print('Enter :q to quit')

    c = input("Enter Molecular Formula in given format - e.g. (P_Cl*5):\t")
    if c.lower() == ":q":
        welcome()
    else:
        q = int(input("Enter Charge on Compound: \t"))

    final_sol = mech(c,q)


    mech(c,q)
    print ("Hybridization:", final_sol[0])
    print("Geometry:", final_sol[1])
    print("Shape:", final_sol[2])
    print("Skeletal Structure:", structure(final_sol[0], int(final_sol[3])), sep="\n")
    choice = input('\n Want to run the program again? (Enter Y or N)\t')
    if choice.lower() == "y":
##        mech(c,q)
        calculator()
    else:
        welcome()


def quiz():
    import random

    def comment (result):
        if result == 100:
            print("You Aced It!")
        elif result >= 75:
            print("That's Impressive!")
        elif result >= 30:
            print("You can always get better!")
        else:
            print("Better Luck Next Time!")

    cont = True
    score = 0
    total = 0
    mol = [
    ["Be_Cl*2", 0],
    ["Be_F*2", 0],
    ["Be_Br*2", 0],
    ["C_O*2", 0],
```

```
["Si_O*2", 0],
["N_O*2", 1],
["B_F*3", 0],
["B_Cl*3", 0],
["B_Br*3", 0],
["C_O*3", -2],
["N_H*3", 0],
["C_H*4", 0],
["N_H*4", 1],
["H*2_O", 0],
["H*3_O", 1],
["P_Cl*5", 0],
["P_Cl*3", 0],
["Cl_F*3", 0],
["I*3", -1],
["S_F*6", 0],
["Xe_F*4", 0]
]
while cont == True:

    quiz_mode = [0,1]
    q_type = quiz_mode[random.randint(0, len(quiz_mode)-1)]

    if q_type == 0:

        if mol == []:
            print("You've Reached the End!")
            print("Your Scored", score, "out of", total)
            print("That's about", str((score/total)*100)[:4] + "%" )
            comment((score/total)*100)
            cont = False
        else:
            total += 3
            chosen = mol[random.randint(0, len(mol)-1)]
            comp = chosen[0]
            charge = chosen[1]
            mol.remove(chosen)
            mech(comp, charge)

            print("The Compound is:", comp, "charge:", charge)
            user_hyb = input("What is the Hybridization of Central Atom?:\t" )
            if user_hyb.lower() == final_sol[0].lower():
                print('✅')
                score += 1
            else:
                print('❎')
            user_geo = input("What is the Geometry?:\t" )
            if user_geo.lower() == final_sol[1].lower():
                print('✅')
                score += 1
            else:
                print('❎')
            user_shape = input("What is the Shape?:\t" )
            if user_shape.lower() == final_sol[2].lower():
                print('✅')
                score += 1
            else:
                print('❎')

            choice = input('\n Want to play again? (Enter Y or N)\t')
            if choice.lower() == "y":
                cont = True
            else:
                print("You Scored", score, "out of", total)
                print("That's about", str((score/total)*100)[:4] + "%" )
                comment((score/total)*100)
                welcome()

    elif q_type == 1:

            total+=1
            valid_hyb = ["sp", "sp2", "sp3", "sp3d", "sp3d2"]
```

```
                hyb = valid_hyb[random.randint(0, len(valid_hyb)-1)]


                if len(valid_hyb) == 0:
                    quiz_mode.pop()

                print(skeletal_str[hyb][0])
                user_h = input("What is the Hybridization of Central Atom in the Structure Shown Above?:\t").lower()

                if valid_hyb.count(user_h) == 0:
                    print("Please Enter Valid Hybridization")
                else:
                    if user_h == hyb:
                        print('✅')
                        score += 1
                    else:
                        print('❎')
                valid_hyb.remove(hyb)
                choice = input('\n Want to play again? (Enter Y or N)\t')
                if choice.lower() == "y":
                    cont = True
                else:
                    print("You Scored", score, "out of", total)
                    print("That's about", str((score/total)*100)[:4] + "%" )
                    comment((score/total)*100)
                    welcome()
welcome()
```

Output

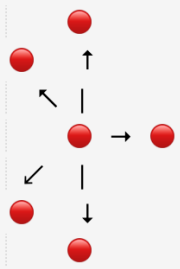
Welcome Screen

```
VALANCE SHELL ELECTRON PAIR REPULSION THEORY ~ VSEPR
~By Hardik Mirg
Select Application Mode:

    Enter (1) for VSEPR Calculator
    Enter (2) for VSEPR Quiz
    Enter (3) for Hybridized Skeletal Structures
    Enter (4) for Periodic Table


Enter mode:
```

MODE 1: VSEPR CALCULATOR

On choosing VSEPR Calculator, entering PCl5 as an example,
We are provided with the **hybridization, Geometry, Shape and Correct Structure of the Compound**

```
Enter mode: 1
Enter :q to quit
Enter Molecular Formula in given format - e.g. (P_Cl*5):    P_Cl*5
Enter Charge on Compound:   0
Hybridization: sp3d
Geometry: Trigonal Bipyramidal
Shape: Regular
Skeletal Structure:
```



```
Want to run the program again? (Enter Y or N)
```

```
Enter Molecular Formula in given format - e.g. (P_Cl*5):    I*3
Enter Charge on Compound:   -1
Hybridization: sp3d
Geometry: Trigonal Bipyramidal
Shape: Linear
Skeletal Structure:

  🔴
  ↑
  |
  🔴
  |
  ↓
  🔴
```

MODE 2: QUIZ MODE

```
Enter mode: 2
The Compound is: N_H*4 charge: 1
What is the Hybridization of Central Atom?: sp3
✅
What is the Geometry?:  tetrahedral
✅
What is the Shape?: pyramidal
❎
Want to play again? (Enter Y or N)  y
      🔴
   🔴  ↑
     ↖ |
      🔴 → 🔴
   ↙  |
   🔴  ↓
      🔴


What is the Hybridization of Central Atom in the Structure Shown Above?:sp3d
Above?:sp3d
```

The quiz mode has two type of questions:

1. The program provides a compound and asks questions about it
2. The program draws the structure and you have to comment on its hybridization

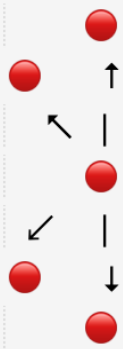
Your Score is shown at the end of the quiz

```
 Want to play again? (Enter Y or N) n
 You Scored 3 out of 4
That's about 75.0%
That's Impressive!
```

MODE 3: STRUCTURE MODE

In the structure mode you can enter the hybridization and no. of lone pairs and you will be provided with the structure

```
Enter mode: 3
Enter Hybridization of the Structure: sp3d
Enter number of Lone Pairs1
Skeletal Structure:
```



MODE 4: THE PERIODIC TABLE

This mode showcases the **Modern Periodic Table** which played an essential role in the functionality of this program as all calculations made to determine the correct structure and shape were done using the **Periodic Properties of the Elements**

```
Enter mode: 4
   H                                                                 He
   Li Be                                          B    C    N  O   F   Ne
   Na Mg                                          Al   Si   P  S   Cl  Ar
   K  Ca Sc Ti  V  Cr Mn  Fe  Co  Ni  Cu Zn  Ga   Ge  As  Se  Br  Kr
   Rb Sr  Y Zr  Nb Mo Tc  Ru  Rh  Pd  Ag Cd  In   Sn  Sb  Te  I   Xe
   Cs Ba  | Hf  Ta W  Re  Os  Ir  Pt  Au Hg  Tl   Pb  Bi  Po  At  Rn
   Fr Ra  | Rf  Db Sg Bh  Hs  Mt  Ds  Rg Cn  Uut Uuq Uup Uuh Uus Uuo
          |


          ↳ La Ce  Pr Nd Pm  Sm Eu Gd Tb Dy  Ho  Er   Tm   Yb    Lu
          ↳ Ac Th  Pa U  Np  Pu Am Cm Bk Cf  Es  Fm   Md   No    Lr
```

Limitations

- The Program cannot handle molecules that contain more than two types of atoms
- Molecules with more than one central atom aren't supported
- The Program cannot show structures of molecules having hybridization higher than sp3d2 as it is really difficult to draw 3d structures on a 2d Screen

Bibliography

- Computer Science textbook
- NCERT Chemistry textbook