

Python기반 미시적 Orbital 구현 자동화 기법 (수소원자 오비탈 경우를 바탕으로)

물리화학2와 대화형프로그래밍 강의페어링
화학과 201721276 전효남 / 오하영, 이진희 교수님 지도

목적

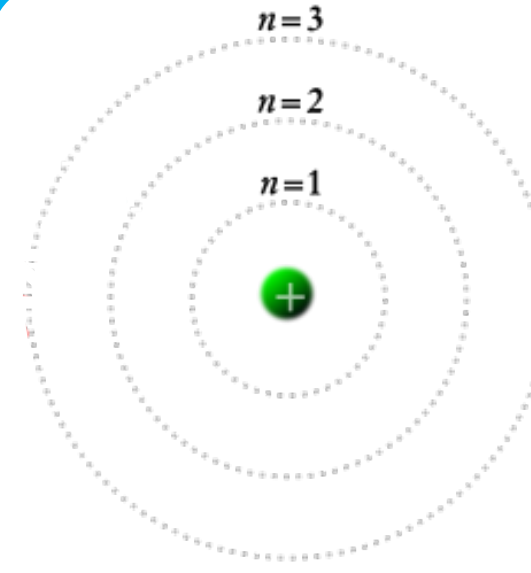
프로그래밍을 통해서 복잡한 슈뢰딩거 방정식을 풀고,
수소의 원자 궤도 함수(오비탈)을 그래픽 모듈로 구현한다.

배경지식-오비탈이란?

원자는 원자핵과 전자로 이루어져 있다. 그 전자가 있는 **전자껍질을 원자 궤도 또는 원자 오비탈**이라고 한다. 원자 궤도란 원자핵 주변의 특정한 영역에서 **원자의 모든 전자들을 찾을 확률**을 의미한다.

전자들은 자신이 가지고 있는 **에너지와 운동량에 따라** 원자 주위에 다른 확률로 존재한다. 예를 들어 **에너지가 높은 전자는 원자로부터 멀리 존재할 확률이 높다**. 또한 전자가 특정한 방향으로의 **운동량을 가지고 있다면** 등근모양이 아닌 **아령과 같은 모양**으로 궤도가 형성될 수 있다.

배경지식



양자수 : 원자 안에서의 **전자의 에너지, 각운동량, 방향** 또는 스핀 등에 대한 정보를 나타내는 수

● **주 양자수**(principal quantum number)(n) : 원자의 전자껍질 또는 원자가 가지는 **에너지준위**를 나타낸다. 일반적으로 주양자수는 n 으로 나타내어지는데 이때 n 의 값은 1부터 k 까지의 **자연수**이다.

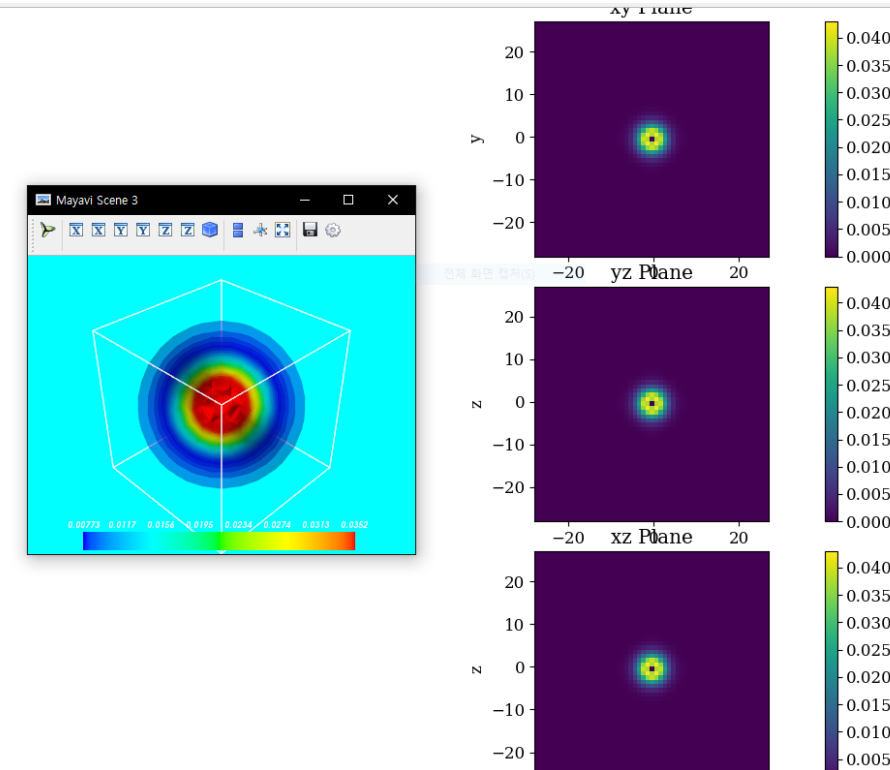
● **방위 양자수**(azimuthal quantum number)(l) : 각양자수라고도 불린다. 오비탈의 **각운동량**을 나타낸다.

0부터 $n-1$ 까지 정수로 나타내며, $l=0$: **s오비탈**, $l=1$: **p오비탈**, $l=2$: **d오비탈** 등의 이름이 붙어 있다. **오비탈의 모양에 영향**을 미치는 양자수이다.

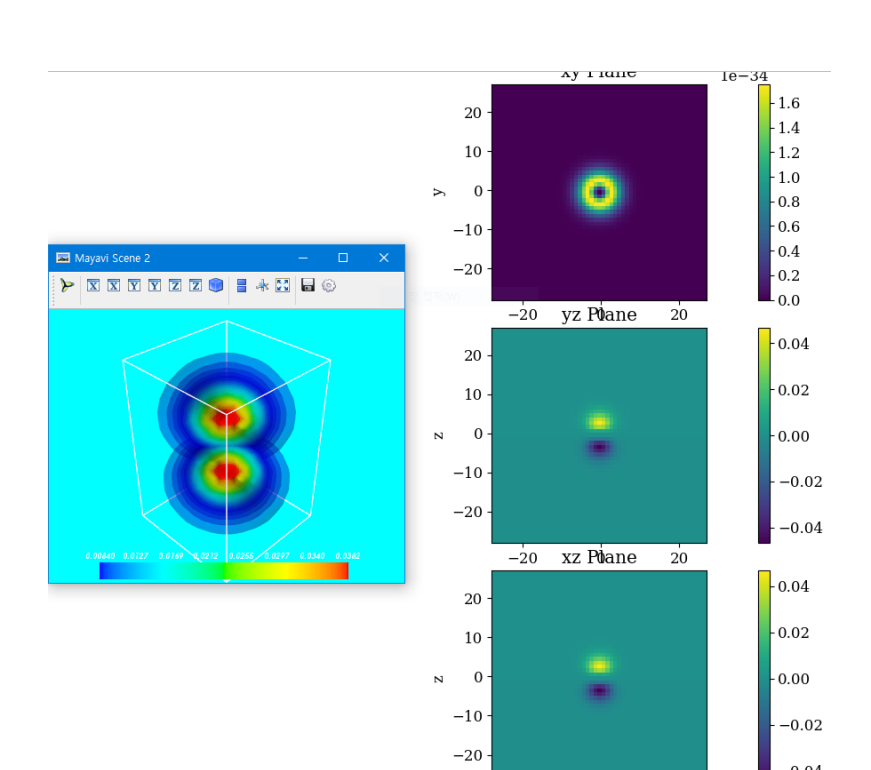
● **자기 양자수**(magnetic quantum number)(m) : 전자 구름이 좌표계에서 어떤 **방향**으로 위치해있는지 알려주는 양자수이다. m 의 범위는 $-l \sim +l$ 까지의 정수로 표현된다.

출력 결과 및 분석

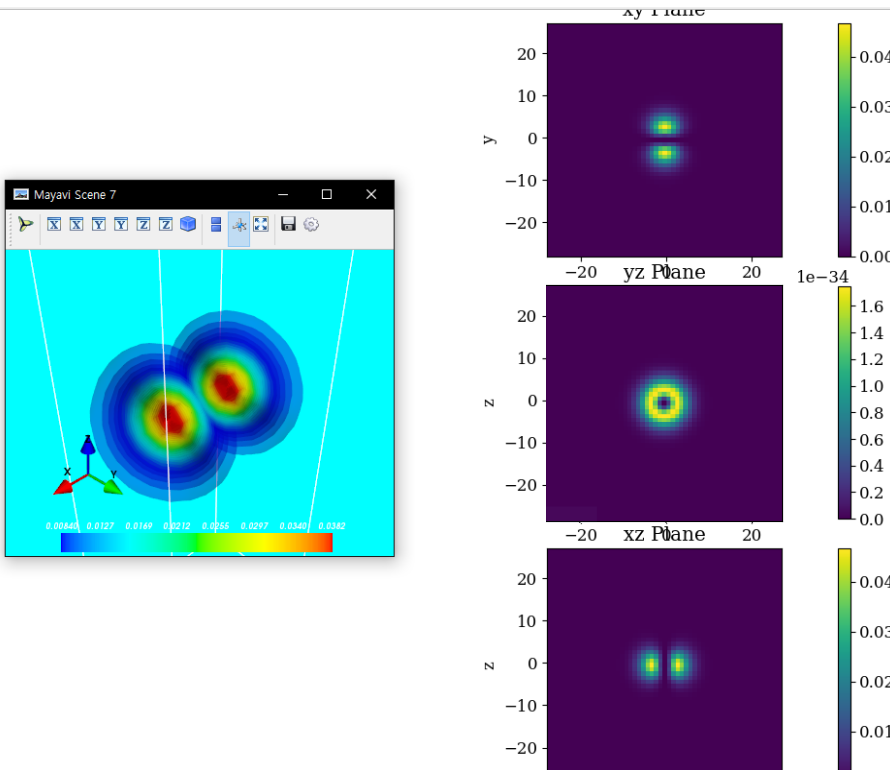
$n=1, l=0, m=0$ 1s오비탈



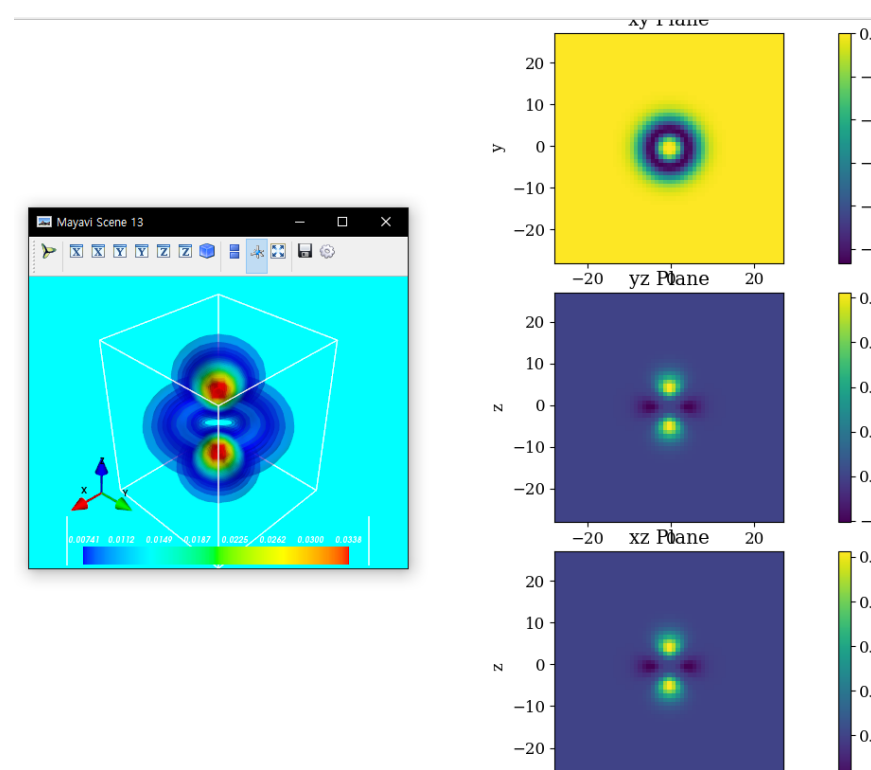
$n=2, l=1, m=0$ 2pz오비탈



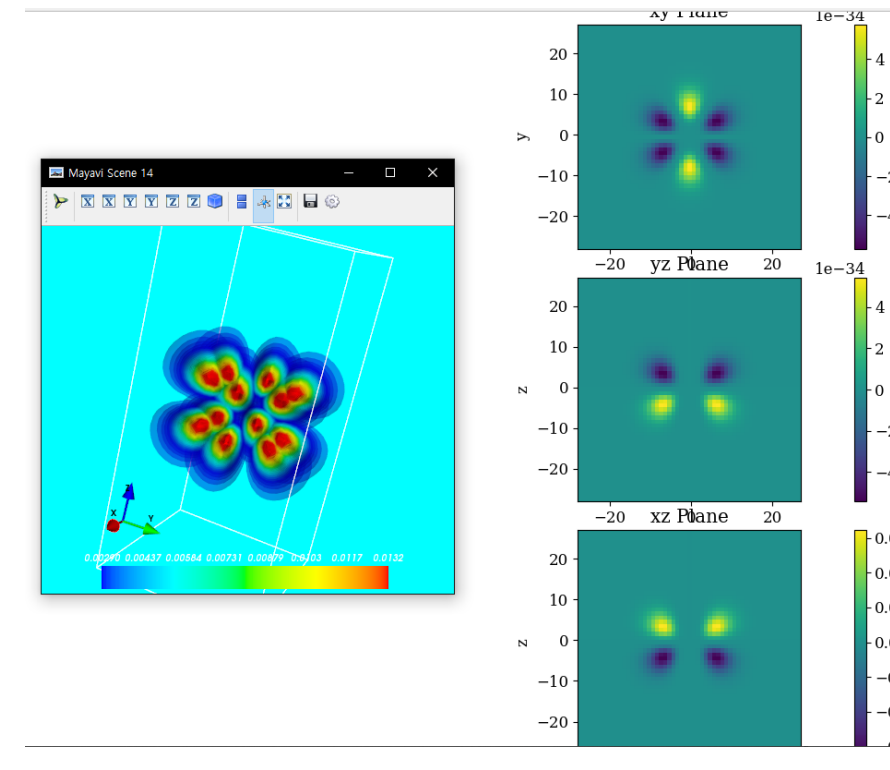
$n=2, l=1, m=1$ 2px오비탈



$n=3, l=2, m=0$ 1dzz오비탈



$n=5, l=4, m=3$



$n=2, l=1, m=1$ 텍스트 출력 결과

```
In [1]: %run "D:\OneDrive - 이진희\OneDrive\Programming\강의페어링\orbital.py"
Python version = 2.7.13 [Enthought, Inc. (x86_64)] (default, Mar 2 2017, 16:05:12) [MSC v.1500 64
Machine readable Python version = sys.version_info(major=2, minor=7, micro=13, releaselevel='final')
The library version numbers used in this notebook are:
mayavi version: 4.5.0
numpy version: 1.13.3
sympy version: 1.0
n must be greater or equal to 1.
Enter a value for n(principal quantum number) n = 2
Enter a value for l(azimuthal quantum number) such that it is less than n (default l=n-1) l = 1
The next value, m, determines which specific orbital will be displayed.
Enter a value(s) for m such that -1 <= m <= 1.
Enter a value for m(magnetic quantum number), (default m=1) m = 0
Electron orbitals are traditionally displayed at a 95% confidence interval.
The isosurface you would see is at a P value that depends on the principal
quantum number and the shape of the orbital. They are analogous to the height
on a normal probability density function plot.
$P = \frac{0.03125}{r^4} \{ \frac{1}{4} \} e^{-1.0 r} \cos^2(\frac{1}{2} \theta)$
Summary
The quantum numbers in this picture are as follows.
n = 2
l = 1
m = 0
The Probability Equations of the following picture is as follows.
$P = \frac{0.03125}{r^4} \{ \frac{1}{4} \} e^{-1.0 r} \cos^2(\frac{1}{2} \theta)$
```

오비탈의 이름 결정법: l 값에 따라 $l=0 \rightarrow s, 1 \rightarrow p, 2 \rightarrow d, 3 \rightarrow f, \dots$ 의 이름이 붙으며, m 값에 따라 방향이 결정된다. 맨 앞의 숫자는 n 의 값이다.
Ex) $2p_x \rightarrow (2=n), (p \rightarrow l=1), (x \rightarrow m\text{값에 의해 결정되는 오비탈의 방향})$

양자수 n 값에 따라 오비탈의 크기가 증가함을 볼 수 있다.(오비탈 단면을 통해 명확히 확인 가능)
양자수 l 값에 따라 오비탈의 모양이 변하는 것을 볼 수 있다. ($l=0$ (s오비탈)은 구형, $l=1$ (p오비탈)은 아령모양 등...)
양자수 m 값에 따라 오비탈의 방향이 변하는 것을 볼 수 있다.

장점과 한계

장점 : 기존의 가우시안(Gaussian, 계산화학용 중합 프로그램)이나 오비탈 뷰어(Orbital Viewer, 다양한 오비탈의 모양을 그려주는 프로그램)와 같은 프로그램과 달리 수소원자 하나에만 집중하였고 이로 인해 n, l, m 이외의 변수를 전혀 입력할 필요 없이 간편하게 수소원자오비탈을 볼 수 있다.(오비탈 뷰어는 질량, α, β, γ 과 같은 약간의 변수가 더 필요하다.) 확률밀도함수 계산 결과를 출력해주기 때문에 파동함수와 슈뢰딩거 방정식으로 정말 오비탈이 구현되는가를 확인할 수 있다.

한계 : 수소원자에만 집중했기 때문에 다른 원자 혹은 분자의 오비탈은 관찰이 불가능하고, 가우시안처럼 최적화된 분자의 구조 분자의 회전 또는 진동과 같은 오비탈 이외의 연산 역시 불가능하다.

참고 문헌

Peter Atkins, 『Atkins』 물리화학 10th Edition, 안운선 옮김, 교보문고, 2017, pp.357~369
코드 참조, [https://github.com/damontallen/Orbitals/blob/master/Hydrogen%20Orbitals%20\(Feb%2018,%202014\)%20\(dynamic%20entry\).ipynb](https://github.com/damontallen/Orbitals/blob/master/Hydrogen%20Orbitals%20(Feb%2018,%202014)%20(dynamic%20entry).ipynb)

코드 요약

실제 코드가 아닌 의사코드(해석코드)입니다.
(코드에 중요도에 따라 텍스트 크기 및 굵기 조절이 되어있습니다.)

```
#모듈 인포트
시스템 모듈 불러오기
텍스트 출력 : 파이썬 버전 = 값
파이썬 버전 정보 변수 초기화
만약 파이썬 버전이 2.7.3이 맞다면:
    mayavi 그래픽 모듈 불러오기
    텍스트 출력 : "mayavi 그래픽 모듈 버전 = 값"
아니라면:
    에러처리

numpy, sympy 모듈 불러오기
방정식 입력을 위한 sympy내의 pi, symbols, factorial, exp, sqrt, cos, sin, re, im,
simplify, Abs 등의 수학적 기호 라이브러리 불러오기
함수를 출력하기 위한 sympy내의 init_printing 라이브러리 불러오기
사용할 변수들의 타입 정의
텍스트 출력 : "numpy 모듈 버전 = 값"
텍스트 출력 : "sympy 모듈 버전 = 값"
```

```
#계산 함수 정의
Legendre polynomial(르장드르 다항식) 정의
Spherical harmonics(구면 조화 함수) 정의
Laguerre polynomial() 정의
Radial function() 정의
Wavefunction(파동함수) 정의
Probability density(확률밀도함수) 정의
```

```
#2D 단면 전용 계산 함수 정의
오비탈 2D 단면 모델 계산을 위한 함수 정의
matplotlib.pyplot의 plt 모듈 불러오기
Signed spherical harmonics(+, - 부호를 가지는 구면 조화 함수) 정의
Signed wavefunction(+, - 부호를 가지는 파동함수) 정의
Signed Probability density(+, - 부호를 가지는 확률밀도함수) 정의
확률밀도함수를 문자로 변환해주는 함수 정의
확률 방정식을 LaTeX 문자로 기록
만약 방정식에 에러가 있다면:
    텍스트 출력 : 확률 방정식에 에러가 있습니다.
    에러처리
변환된 문자열을 호출 가능한 함수로 변환
```

```
#변수 정의
구면 좌표계 성분(반지름, 각도) 초기화
```

```
#출력 가능 함수 정의
오비탈 3D 모델 출력 함수 정의
출력결과 화면 레이아웃 설정
그래픽 모듈 레이아웃 설정(모델의 색 및 배경 색 배열 설정)
양자수 범위 정의( $n \geq 1, 0 \leq l \leq n-1, -l \leq m \leq l$ )
등고선의 갯수를 30개로 초기화(3D 모델의 정확도를 나타냄)
불투명도를 0.5로 초기화(1은 완전 불투명, 0은 거의 투명)
```

```
#입력
주양자수 n 값을 입력 받음
만약 n 값이 1보다 작으면:
    텍스트 출력 : "n은 자연수여야 합니다."
    에러 출력
```

```
부양자수 l 값을 입력 받음
만약 l 값이 입력되지 않는다면:
    l=n-1로 치환
만약 l 값이 0보다 작거나 n보다 같거나 크다면:
    l=n-1로 치환
    텍스트 출력 : "l은 0 <= l <= n 범위에 속할 수 있다. l=n-1로 자동으로 설정한다."
```

```
자기양자수 m 값을 입력 받음
만약 m 값이 입력되지 않는다면:
    m=l로 치환
만약 m > l or m < -l 이라면:
    m=l로 치환
    텍스트 출력 : "m은 -l <= m <= l 범위에 속할 수 있다. m=l로 자동으로 설정한다."
```

```
#설명 출력
텍스트 출력 : "오비탈은 전자 확률 95% 범위 까지 출력한다. 확률밀도 P의 값과 모양이 양자번호에 따라 달라질 수 있다."
```

```
#결과 출력
3D, 2D 오비탈 모델 출력 함수 호출
출력결과(양자수(n,l,m) 값, 확률밀도함수) 요약 출력
LaTeX(파이썬에서 함수를 표현할 수 있는 방법) 문자로 확률 밀도 함수 출력
```