심화 연구 · 탐구 활동 보고서

<table>
<tr><th>주제</th><td>렌즈에서의 스넬의 법칙 적용</td><th>이유(동기)</th><td>아래 서술</td></tr>
<tr><th>주요 내용 요약</th><td colspan="3">스넬의 법칙을 이용하면 빛이 렌즈를 통과할 때 굴절각을 구하는 식을 유도할 수 있다. 이를 파이썬(Python) 언어를 통해 구현하여 빛이 렌즈를 통과할 때의 형태를 관찰할 수 있다.</td></tr>
<tr><th>융합 과목명 및 연계내용</th><td colspan="3">물리학 1-파동과 정보 통신-파동의 성질, 전반사와 광통신</td></tr>
<tr><th>활동 방법</th><td colspan="3">렌즈의 형태와 빛의 궤적을 함수로써 생각한 후, 빛의 기울기 변화를 계산하는 공식을 유도한 후, 컴퓨터를 통해 구현한다.</td></tr>
<tr><th colspan="4">조사 내용 및 알게된 점</th></tr>
</table>

동기 : 스넬의 법칙에 관한 내용을 배우면서, 매질 1, 2가 평평한 상태로 맞닿아 있는 경우에서만 다루고, 렌즈의 굴절에 대해서는 간단하게만 배운다. 따라서 나는 같은 굴절의 성질을 이용하는 렌즈에서는 어떤 방식으로 굴절이 일어나는 것인지 궁금하였고, 이를 알아보기 위해 탐구를 진행하였다.

우선, 렌즈와 빛을 수식으로 나타내기 위해, 렌즈와 빛을 함수식으로 표현한다. 볼록렌즈와 오목렌즈 두가지 경우에 대해 식이 존재하지만, 유도하는 방법은 같으므로 여기서는 볼록렌즈에 대한 식만 유도한다. 렌즈는 최고차항의 계수의 부호가 반대인 두 이차함수를 이용해 나타낸다.

아랫부분 : $f(x) = \frac{1}{10}x^2 \quad (-5 \le x \le 5)$

윗부분 : $g(x) = -\frac{1}{10}x^2 + 5 \quad (-5 \le x \le 5)$

그다음은 빛의 기울기를 구하는 식을 유도한 후, 함수로 나타낸다. 아래 [그림1] 에 있는 증명에 따라,

$$m1 = \tan(\arcsin(\frac{2}{3}\sin(\frac{\pi}{2} + \arctan(\frac{1}{f'(a)}))) - \arctan(\frac{1}{f'(a)}))$$

$$m2 = -\tan(\arcsin(\frac{3}{2}\sin(\arctan(\frac{1}{g'(\alpha)}) + \arctan(m1))) + \arctan(\frac{1}{g'(\alpha)}))$$

$$L(x) = \begin{cases} a & (L(x) < f(a)) \\ m1(x-a) + f(a) & (f(a) < L(x) < g(\alpha)) \\ m2(x-\alpha) + g(\alpha) & (L(x) > g(\alpha)) \end{cases}$$

와 같이 빛의 y좌표에 따른 궤적을 함수로 나타낼 수 있다. a는 렌즈에 도달하기 전의 빛의 x좌표이고, α는 f(x)와 만난 후 굴절된 빛의 그래프가 g(x)와 만나는 점을 근의 공식을 이용하여 구한 두 근 중 더 큰 값을 의미한다. 빛의 그래프와 g(x) 그래프가 만나는 점을 생각해 보았을 때 y 값이 더 큰 값이 실제로 빛이 렌즈의 윗부분과 만나는 점이기 때문이다. m1은 x축에 대해 수직으로 진행하던 빛이 렌즈와 처음 만난 후 굴절한 기울기를 나타내고, m2는 1차로 굴절한 빛이 렌즈의 윗부분과 만나 다시 한번 굴절한 후의 기울기를 나타낸다. 이렇게 도출한 식을 바탕으로 파이썬(Python)을 이용해 구현하면 실제 렌즈에서 빛이 굴절이 어떻게 일어나는지 알아볼 수 있다. 제작한 프로그램을 실행하면 [그림2] 와 같은 결과가 나타난다.

결과를 통해 볼록렌즈에서는 빛이 모이고, 오목렌즈에서는 분산된다는 것을 확인할 수 있다. 추가로, 볼록렌즈의 가장자리 부분에서는 빛의 입사각이 임계각을 넘어 전반사가 일어나 빛이 굴절되지 못하여 중간에 빛이 끊긴 것처럼 보이는 부분이 있다. 렌즈의 전반사에 대해서는 들어본 적이 없으므로 인터넷을 통해 추가로 조사하여 보았고, 실제로 렌즈의 굴곡에 따라 전반사 때문에 보이지 않는 영역이 생기며 현미경의 경우 전반사로 인해 손실되는 부분을 줄이기 위해 대물렌즈 사이에 물이나 기름을 채운다는 것을 알게 되었다.

또한, 이번 탐구에서는 굴절각을 알아내기 위해 삼각비를 이용하여 식을 유도하였지만, 추가로 조사한 결과, 실제 광학 설계 및 컴퓨터 그래픽에서는 위의 방법 대신, 2x2 행렬을 이용한 계산을 이용한다는 것을 알았고, 나중에 이를 적용하여 추가로 활동을 해보고 싶다고 생각하였다.

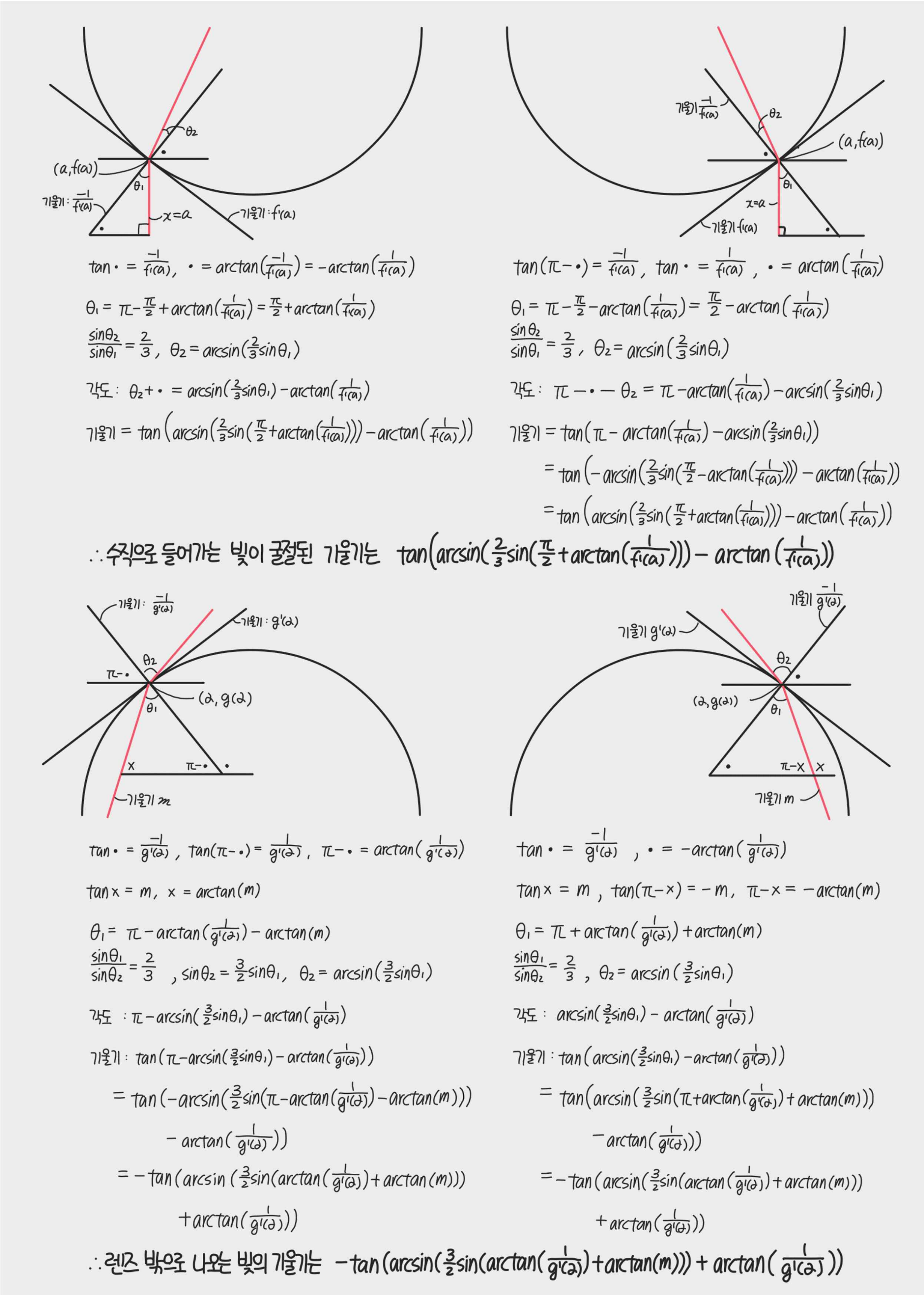

그림 1 굴절한 빛의 기울기 유도

결과 예시:

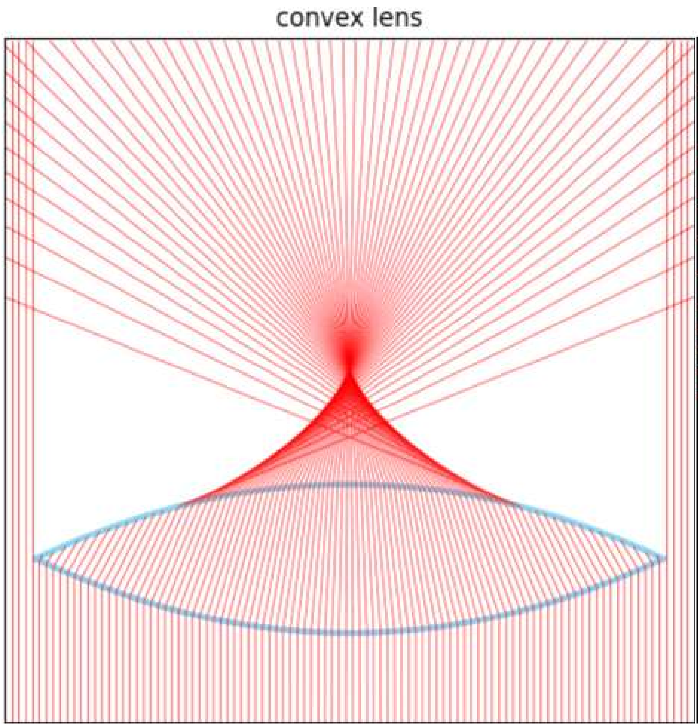

그림 2 볼록렌즈

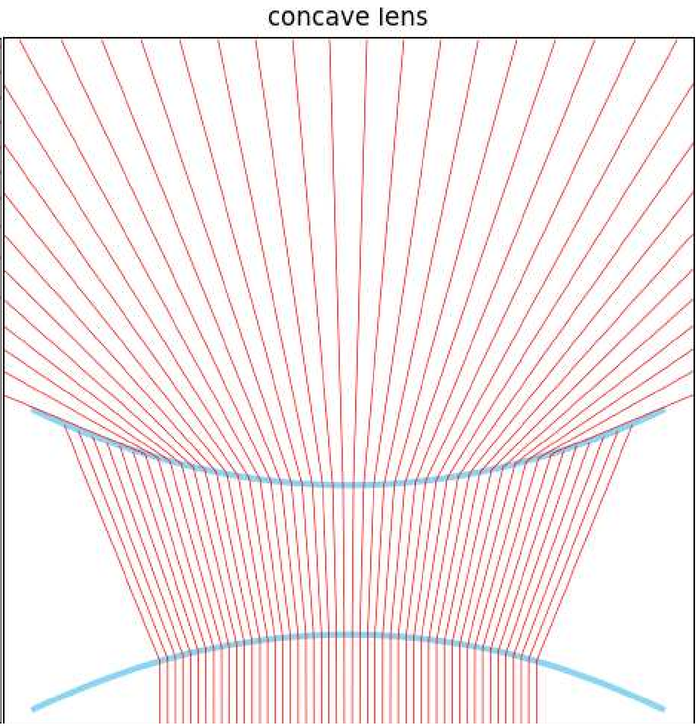

그림 3 오목렌즈

실제 작성 코드:

```
import matplotlib.pyplot as plt
import numpy as np

globalcontact

def f(x):
 return 1/10*x**2
def f_d(x):
 return 1/5*x

def g(x):
 return-1/10*x**2+5
def g_d(x):
 return-1/5*x

def M1(x):
 return np.tan(np.pi/2-np.arctan(-f_d(x))+np.arcsin(2/3*np.sin(np.arctan(-f_d(x)))))
def M2(x):
 return-np.tan(np.arcsin(3/2*np.sin(np.arctan(1/g_d(contact))+np.arctan(M1(x))))+np.arctan
(1/g_d(contact)))

def find_x(x, y):
 if(y<=f(x) or x<-5 or x>5):
  return x
 elif(f(x)<y<=g(contact)):
  return 1/M1(x)*(y-f(x))+x
 else:
  return 1/M2(x)*(y-g(contact))+contact
```

```python
lens_d= [list(np.linspace(-5, 5, 100)), list(f(np.linspace(-5, 5, 100)))]
lens_u= [list(np.linspace(-5, 5, 100)), list(g(np.linspace(-5, 5, 100)))]
ray_x, ray_y= [], []

plt.figure(figsize=(6, 6))
plt.axis([-5.5, 5.5, -3, 20])
plt.xticks(range(0), range(0))
plt.yticks(range(0), range(0))
plt.title("convex lens")
plt.plot(lens_d[0], lens_d[1], color='skyblue', linewidth='3', zorder=0)
plt.plot(lens_u[0], lens_u[1], color='skyblue', linewidth='3', zorder=0)

for x in np.linspace(-5.5, 5.5, 100):
 alpha= -5*M1(x)+np.sqrt(25*M1(x)**2+10*M1(x)*x-10*f(x)+50)
 beta= -5*M1(x)-np.sqrt(25*M1(x)**2+10*M1(x)*x-10*f(x)+50)
 contactalpha if g(alpha)>g(beta) else beta
 for y in np.arange(-3, 20, 0.1):
  ray_x.append(find_x(x, y))
  ray_y.append(y)
 plt.plot(ray_x, ray_y, color='red', linewidth='0.5', zorder=1)
 ray_x.clear()
 ray_y.clear()

plt.show()
```

참고 문헌	**물리학1 교과서** **https://en.wikipedia.org/wiki/Ray_transfer_matrix_analysis**