텍스트, 클립아트이(가) 표시된 사진

자동 생성된 설명

코사인의 최대값 구하기

회전 타원체의 표면적과 근사화

결과와 코드 작성 과정



­­

|  |
| --- |
| 과목명: 시뮬레이션 기초 및 실습 |
| 교수: 김지범 |
| 학과: 컴퓨터공학부 |
| 학번: 202201479 |
| 이름: 박지원 |
| 제출일: 2023년 03월 19일 |

**목차**

**Ⅰ. 문제 1번**

**ⅰ. 결과 스크린 샷**

**ⅱ. 코드 작성 과정**

**Ⅱ. 문제 2번**

**ⅰ. 결과 스크린 샷**

**ⅱ. 코드 작성 과정**

**Ⅲ. 마무리**

**Ⅰ. 문제 1번 \_ ⅰ. 결과 스크린 샷**



텍스트, 편지이(가) 표시된 사진

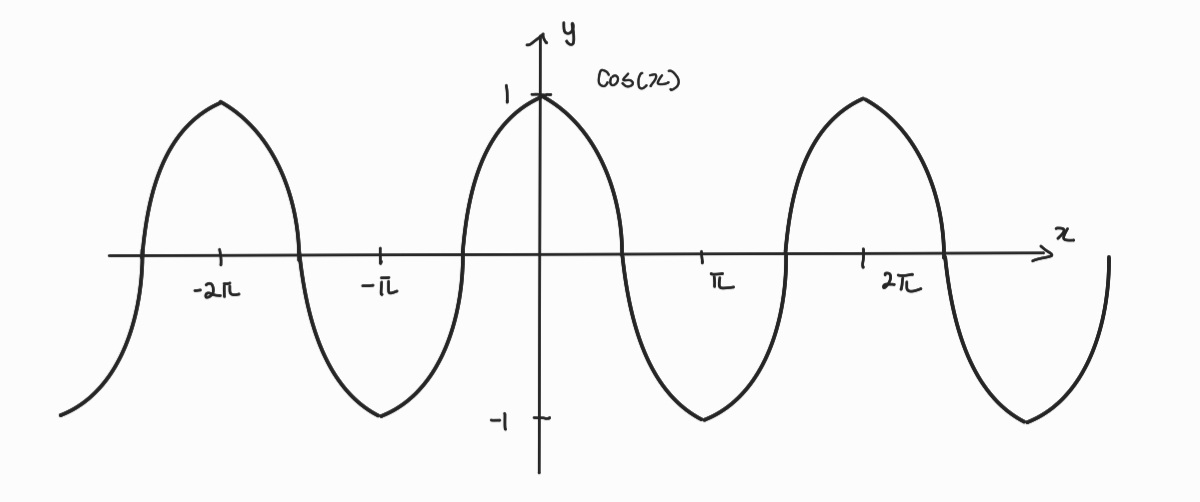
자동 생성된 설명텍스트, 편지이(가) 표시된 사진

자동 생성된 설명



**Ⅰ. 문제 1번 \_ ⅱ. 코드 작성 과정**

우선, 범위인 L과 R을 입력하기 위해 input 함수를 이용하였다. 그리고 직접 확인하기 편하도록 cos(L)과 cos(R)값을 순서대로 disp 함수를 이용하여 출력하였다.



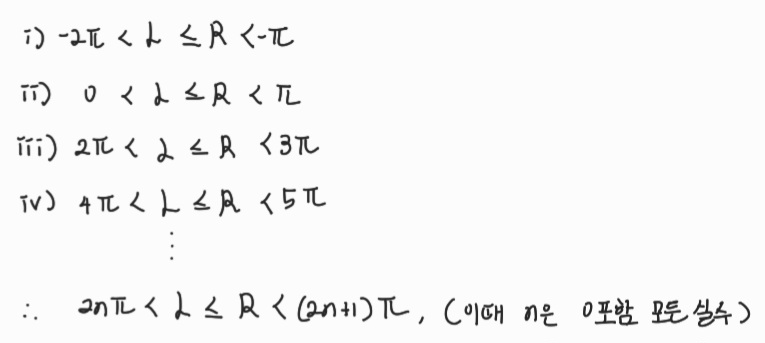
위의 그림은 cos(x)의 그래프를 나타낸 것이다. [L, R]의 범위에 따른 최댓값을 구해야 하는 문제이기 때문에 우선 크게 3가지의 분기로 나누었다. 첫 번째는 1이 아닌 cos(L)이 최댓값이 되는 경우, 두 번째는 1이 아닌 cos(R)이 최댓값이 되는 경우, 마지막으로는 최댓값이 무조건 1이 되는 경우이다. 위의 순서대로 과정을 서술하겠다.

1. 1이 아닌 cos(L)이 최댓값이 되는 경우

도표이(가) 표시된 사진

자동 생성된 설명

cos(L)이 최댓값이 되는 경우는 그래프를 보았을 때, 빨간색으로 표시한 구간 내에서이다.



빨간 색으로 표시된 부분들을 쭉 적다 보면 2\*n\*pi < L <= R < (2\*n+1)\*pi,(이때 n은 0을 포함한 모든 실수) 의 형태를 반복하며 주기를 보인다는 것을 알 수 있다. 이제 이 범위를 코드로 표현해보도록 하겠다. 범위를 나타내기 위해 if함수로 분기를 나누었다.



즉 위의 코드로 표현하였는데, && 연산자를 기준으로 구분하여 서술하도록 하겠다. 서술하기에 앞서 if와 elseif의 조건 속에는 음수 범위까지 포함하 여야만 한다. 따라서 abs 함수를 이용하여 L과 R 모두 절대값으로 표현함을 인지하고 서술하였다.



2\*n\*pi < L <= R < (2\*n+1)\*pi 의 형태를 보면 L을 2\*pi로 나누었을 때의 몫이 R을 2\*pi로 나누었을 때 보다 작거나 같아 야만 빨간 색으로 표시된 부분 내에 속할 수 있다.

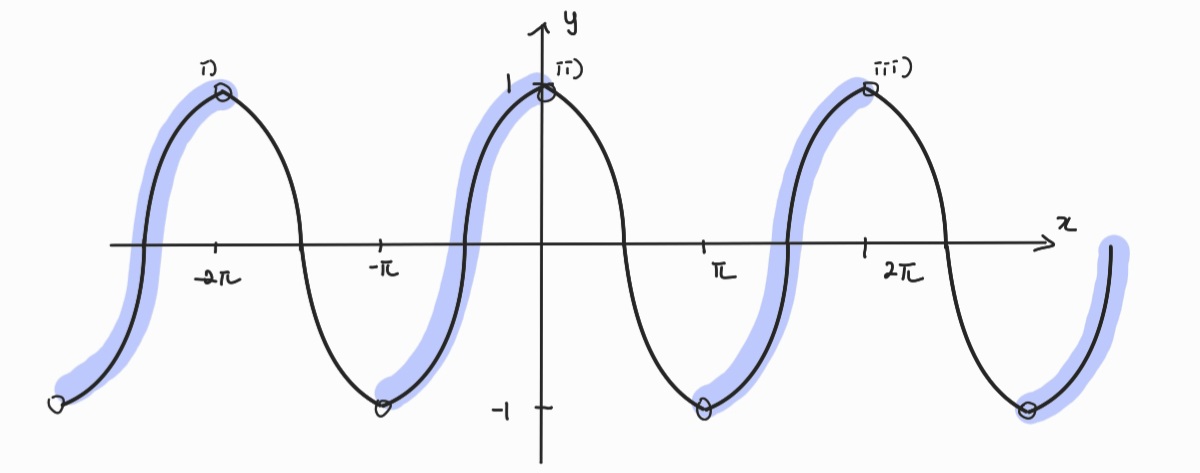


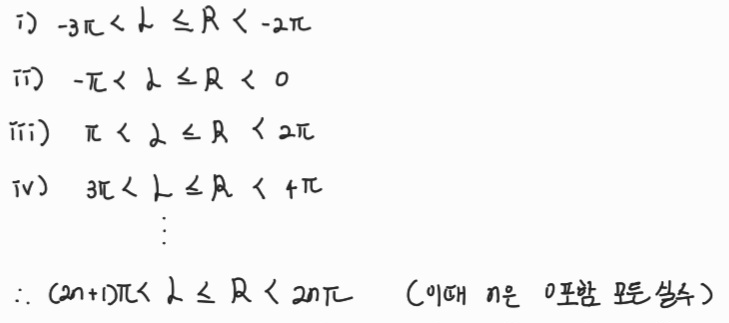
이때, L을 2\*pi로 나눈 몫이 0이 되어버리면, 2\*n\*pi = L이 되어버린다. 따라서 최댓값이 1이 되어버리기 때문에 L은 2\*pi로 나눈 몫이 0이 되어서는 안된다.



마지막으로 서술할 부분은, 만일 [L, R]의 간격이 2 혹은 2\*pi 이상이 되어버리면 중간에 2\*n\*pi가 되는 부분을 꼈다는 뜻이고, 그러면 최댓값이 1이 되어버린다. 따라서 [L, R]의 간격은 2 미만, 2\*pi 미만이어야만 한다.

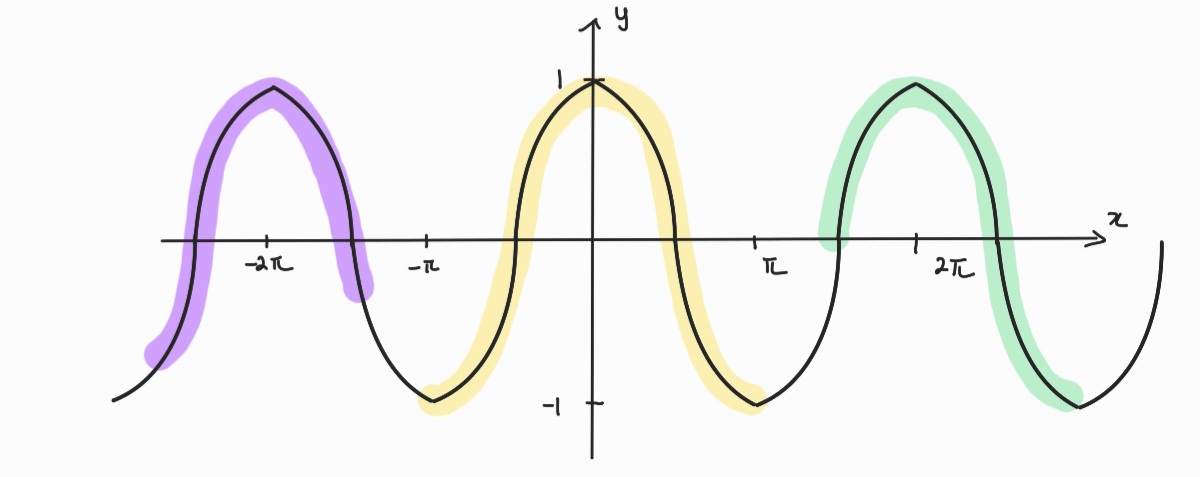
위의 코드를 통해 최댓값이 1이 아닌 cos(L)의 경우를 표현하였다. 위의 조건을 충족시킨다면, cos(L)을 fprintf 함수를 이용하여 출력하도록 하였다.

1. 1이 아닌 cos(L)이 최댓값이 되는 경우

cos(R)이 최댓값이 되는 경우는 그래프를 보았을 때, 파란색으로 표시한 구간 내에서이다.

파란 색으로 표시된 부분들 또한, (2\*n+1)\*pi < L <= R < 2\*n\*pi,(이때 n은 0을 포함한 모든 실수) 의 형태를 반복하며 주기를 보인다는 것을 알 수 있다. 따라서 위에서 보인 cos(L)의 if 조건 부분을 조금만 고치면 코드를 바로 짤 수 있다.

위에 미리 쓰였던, 1이 아닌 cos(L)이 최댓값인 경우를 if 함수로 썼기 때문에 이번에는 elseif 함수를 이용하였다. 원리는 위에 쓰인 cos(L)이 최댓값일 경우의 조건과 같다. 즉, 앞선 조건과는 반대로 L을 2\*pi로 나눈 몫이 R을 2\*pi로 나눈 몫보다 커야 한다. R이 2\*n\*pi로 표현되면 마찬가지로 최댓값이 1이 되므로, R을 2\*pi로 나눈 몫은 0이어선 안 된다. 그리고 또한 [L, R]의 간격이 2혹은 2\*pi 이상이면, 최댓값이 1이 되는 2\*n\*pi의 주기를 포함하고 있기 때문에 [L, R]의 간격이 2 미만 그리고 2\*pi 미만이어야 함을 조건으로 하였다. 이렇게 1이 아닌 cos(R)이 최댓값이 되는 조건을 코드로 작성하였다. 입력 값이 조건에 따라 참일 경우, fprintf 함수를 통해 결과를 출력하도록 하였다.

1. 최댓값이 무조건 1인 경우

위의 그래프의 보라색 선, 노란 선, 초록 선 등과 같은 부분에 해당하는 경우에는 무조건 절대값이 1이 된다. 즉, 앞선 두 조건에 모두 해당하지 않는 경우이므로 else 함수를 이용해 최댓값이 1임을 fprintf 함수로 출력하였다.

즉, 총 3개의 분기로 나누어 코드를 작성할 수 있다.

**Ⅱ. 문제 2번 \_ ⅰ. 결과 스크린 샷**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

**Ⅱ. 문제 2번 \_ ⅱ. 코드 작성 과정**

우선 적도 반경(r\_1)과 극 반경(r\_2)을 입력을 받아야 하므로 input 함수를 이용하였다. 회전 타원체의 표면적의 식,를 A라는 변수에 저장하였다. 그리고 회전 타원체의 표면적을 근사화 시킨 식, 를 A\_approximation이라는 변수에 저장하였다. 이때 각각의 변수 A와 A\_approximation 전에 명령창에 출력 시 충분한 소수점 자리 수를 볼 수 있도록 format long을 사용하였다. 입력시킨 r\_1과 r\_2를 대입한 A 와 A\_approximation의 값을 fprintf 함수를 이용하여 출력하였다. 이때, format long을 전에 이용하였으므로 출력 값은 %f로 받았다.

**Ⅲ. 마무리**

위에 작성한 대로 과제 문제 1번과 2번을 모두 해결하였다. 생각보다 1번의 범위를 나누는 게 쉽지 않아 애를 먹었다. 맨 처음에 적은 코드에는 오류가 있어 아예 코드를 새로 작성한 것이다. 처음에는 최댓값이 무조건 1이 되는 조건을 맨 처음에 적었지만, 오히려 cos(L)과 cos(R)이 최댓값인 경우와 그 외는 모두 1이라는 코드를 작성하는 게 더 편할 것 같아 접근 방향을 바꾸었다. 하지만 이 또한, else를 제외한 if와 elseif 구문에 안전장치를 모두 달아야만 했기 때문에, 예외적인 상황 없이 조건을 쓰는 게 쉽진 않았다. 하지만 직접 그래프를 그리고, 그림을 통해 범위를 직접 나누며 생각하다 보니 간격의 차가 주기가 되지 않아야 함을 깨달을 수 있었다. 보고서를 작성하기 바로 전에 L = 3\*pi, R = 4\*pi + 0.2를 입력하였을 때, 최댓값이 제대로 나오지 않아 고민이었다. 하지만 보고서 작성하기 이전에 간격에 대한 조건을 잘못 걸었음을 알고, 고쳐서 코드를 제대로 완성시킬 수 있었다. 힌트 속 테스트 케이스 이외의 케이스도 성공하니 굉장히 뿌듯했다.

두번째 문제는 문제가 길어 조금 주춤거렸지만, 생각보다 1번 보다 훨씬 단순한 문제였기에 큰 어려움 없이 풀 수 있었지만 공식을 집어넣을 때 괄호를 제대로 썼는지에 대한 약간의 혼동이 조금 있었다. 사소한 오타를 확인하는 게 중요하다는 걸 느끼게 해준 문제였다.

수업시간에 배운 개념들을 이용해 문제를 스스로 직접 풀어보니, 머리 속에 개념이 확실히 잡힐 수 있던 좋은 경험이었다.