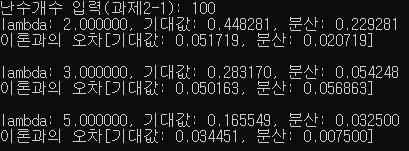
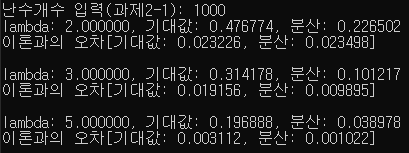
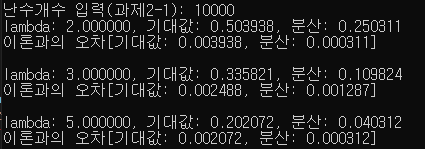
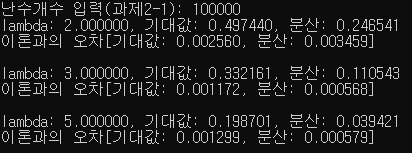
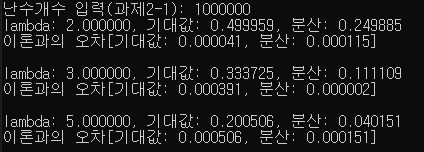
[고소실\_5주차과제]2반\_20161595\_배성현

**과제 2-1**

Bisection 방법을 사용하여 위와 같이 난수의 개수를 늘려가면서 실험해보았을 때, 모든 경우에서 대체적으로 오차가 크지 않아 이론적인 값들과 비슷하게 나온다는 것을 확인 할 수 있었다. 또한 생성한 난수의 수가 많아질수록 이론적인 기대값, 분산과 실험을 통하여 구한 기대값, 분산과의 오차가 점점 작아지는 것을 확인 할 수 있었다.

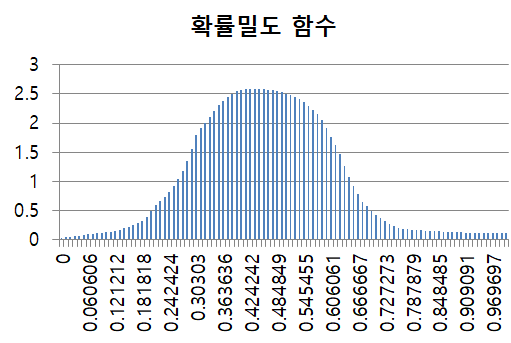
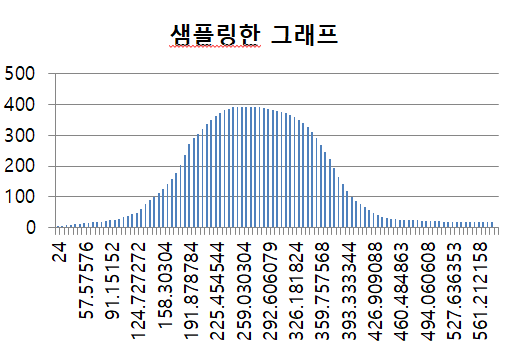


그리고 위와 같이 1000000개 정도의 난수를 생성하였을 때, 시간은 오래 걸리지만 최대 오차 0.0005정도의 값을 가져 이론적인 평균값과 분산값과 충분히 일치하게 된다.

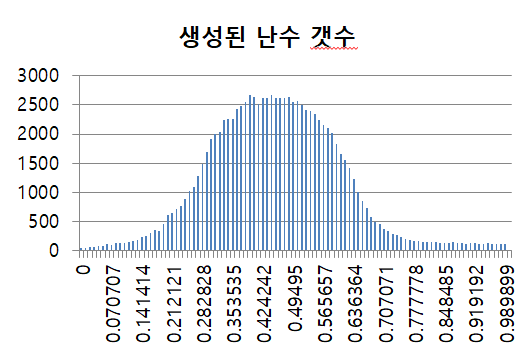


(과제 2-1을 활성화 하기 위해서는 main.cpp에서 program2\_4 주석 제거)

**과제 2-2(ii)**



먼저 CurveSampling프로그램을 이용하여 생성한 데이터를 그래프를 통해 나타내게 되면 왼쪽 위와 같고, 이를 통하여 x의 범위를 [0,1]로 정규화 한 뒤 확률 밀도 함수를 만들어 그래프를 통해 나타내었을 때 오른쪽 위와 같았다. 두 그래프의 형태를 비교하여 보았을 때 샘플링한 데이터를 따라 확률 밀도 함수를 잘 생성하였음을 알 수 있다. 그리고 이를 통하여 누적분포함수를 만들고 (0,1)을 사이의 랜덤하게 생성된 난수 Ui(100000개 생성)를 사용해 fx=Fx(Xi)-Ui=0의 해 Xi를 bisection방법을 이용하여 구하였다. 그리고 이렇게 생성된 Xi들을 각각의 구간(구간의 크기 h)으로 나누고 그 수를 세어 히스토그램 데이터를 구축한 뒤 이를 그래프로 표현하였을 때 아래와 같았다.





fx=Fx(Xi)-Ui=0을 통하여 Xi를 찾기 때문에 Fx의 변화량이 가장 큰 지점에서 Xi난수가 많이 생기게 된다. 즉 확률 밀도 함수의 함숫값이 가장 큰 지점에서 난수가 가장 많이 생성되어야 하고, 위의 생성된 난수 개수의 그래프를 보았을 때 이와 유사하게 나타나는 것을 확인할 수 있다. 또한 위의 그림과 같이 난수 생성으로 만들어진 xi데이터의 분산과 확률 밀도 함수의 분산을 비교하여 보면 두 분산값이 거의 유사하게 나옴을 확인 할 수 있다(프로그램 수행 후 histogram.txt에 저장됨). 따라서 프로그램은 유효하다고 할 수 있다.

**과제 2-2(iii)**

먼저 Newton-Raphson 방법의 초기값을 구하는 방법으로는 [0,1]사이의 중간값을 사용하도록 하였는데 전체 데이터의 개수가 numOfSample이고 균등한 구간으로 쪼개져 있기 때문에, 배열에서 numOfSample/2한 인덱스의 x값을 찾아 이를 초기값으로 사용하였다.



또한 [x0, xn]구간을 초기값으로 하여 Bisection방법을 4회 정도 반복하여 구간의 폭을 줄인 후, 그 중점을 Newton 방법의 초기값으로 사용하는 방법을 사용하였을 때 걸린 시간은 아래와 같다.

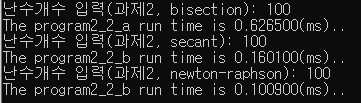


위의 두 결과를 비교하여 보았을 때 중간값을 초기값으로 잡는 것보다, Bisection을 통하여 초기값을 구하는 방법이 조금 더 느린 것(Bisection이 더 효율적이지 않음)을 확인할 수 있다. 또한 위에서는 Bisection방법을 4회정도 반복하여 구간의 폭을 줄였기 때문에, Bisection의 반복을 3회로 바꾸어서도 실험하여 보았고 이 때의 결과는 아래와 같다.

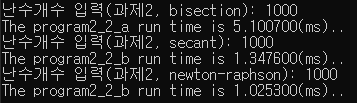


위와 같이 Bisection을 3회 반복하였을 때에는 4회 반복하였을 때보다 시간이 적게 걸린 것을 확인할 수 있었다. 따라서 Newton-Raphson에 Bisection을 사용할 때는 반복 횟수를 잘 조절하여야 한다는 것을 알 수 있다.

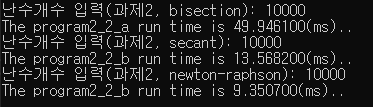
**과제 2-2(iv)**



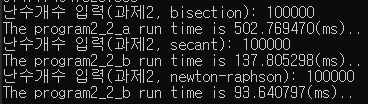
(100개의 난수 생성 시 시간)



(1000개의 난수 생성 시 시간)



(10000개의 난수 생성 시 시간)



(100000개의 난수 생성 시 시간)

위와 같이 100개, 1000개, 10000개, 100000개의 상황에 대하여 난수를 생성하여 bisection, secant, newton-raphson방법을 통하여 실험을 하여 보았고, 위의 사진들을 통해 수치적으로 확인 할 수 있듯이 Newton-Raphson이 가장 빠르고, 그 다음으로 Secant, 그 다음으로 Bisection 순인 것을 알 수 있다.