**고급소프트웨어실습1 - 5주차 과제**

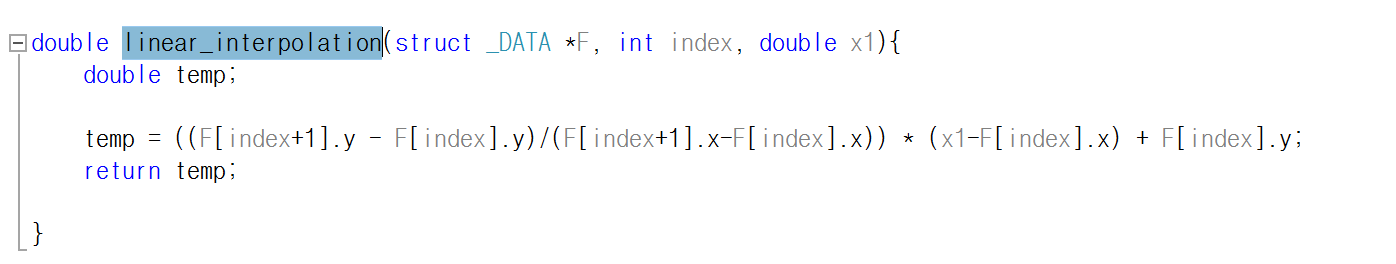
수치 컴퓨팅 실험 2: 특정 확률 사건의 생성

|  |  |  |
| --- | --- | --- |
| **분반** | **:** | 3 |
| **학번** | **:** | 20121635 |
| **이름** | **:** | 장종석 |

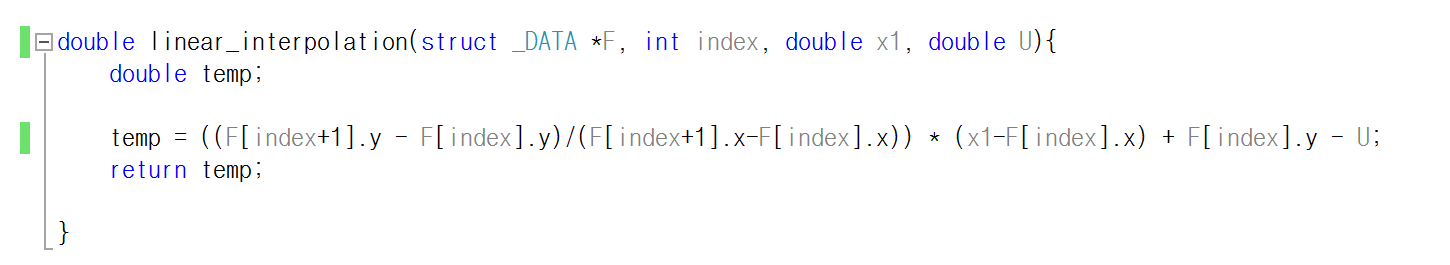
1. ***여러분이 실습 시간에 작성한 코드(프로그램 2-2)를 잘 가다듬어 문제 없이 그리고 효율적으로 수행되도록 수정하라 (프로그램 2-2(a))***

기존 실습해서 작성했던 프로그램 중, 선형변환을 하는 함수를 변환하였다. 선형변환 함수를 사용하는 부분을 확인해보니, 모두 선형변환 후, 임의의 값 U를 빼서 사용하는 것을 확인하였다. 그에 따라, 선형변환 후 값을 계산함과 동시에, 값을 도출하도록 만들었다.

<변경 전 함수>



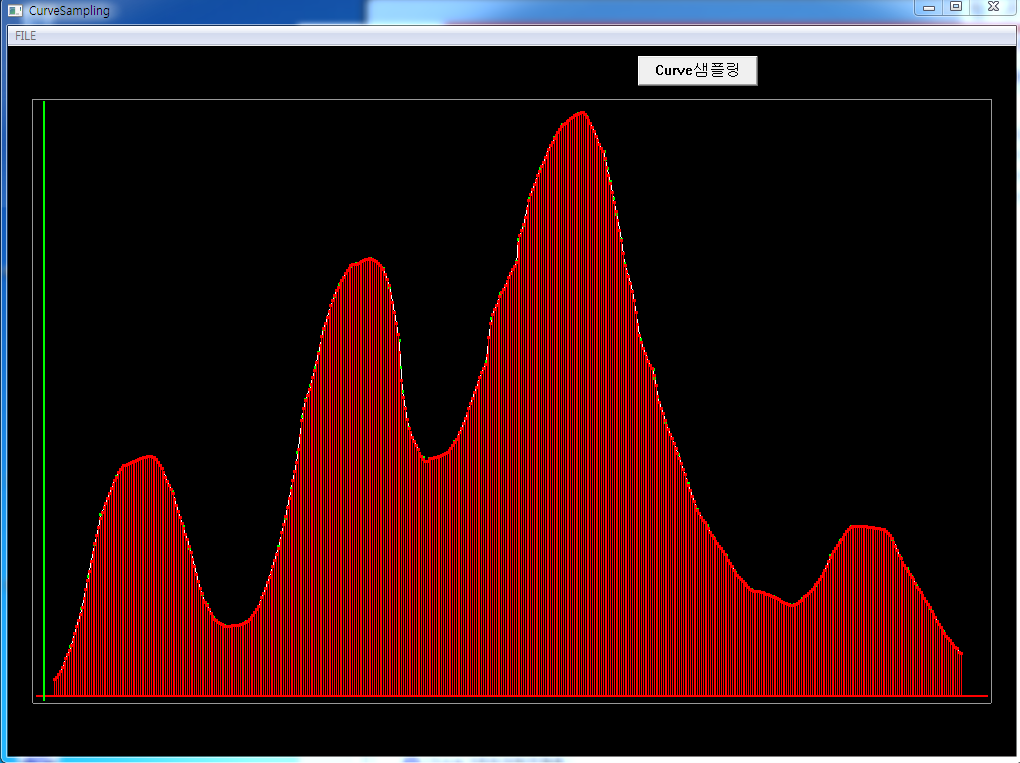
<변경 후 함수>



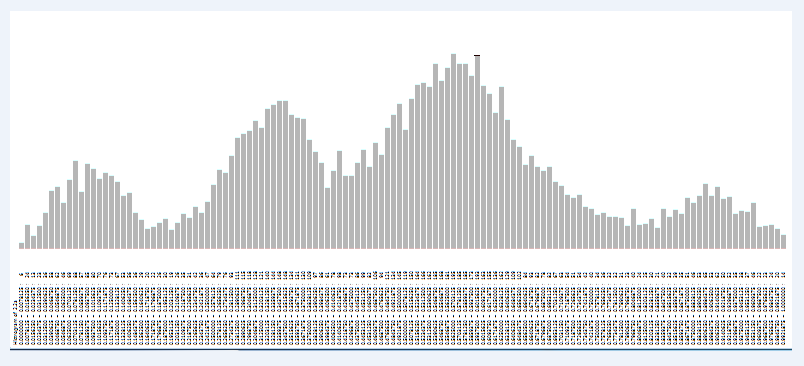
1. ***주어진 확률 밀도 함수에 대하여 자신의 코드가 올바르게 난수를 생성하는지 통계적으로 확인하라. 이를 위하여 아래에서 설명하는 내용의 코드 (프로그램 2-3)을 작성한 후 이를 활용하라.***

먼저 실습에서 난수를 생성하는 방법을 설명하자면, 다음과 같다. 확률 함수의 누적분포 함수를 구하게 되면, 누적분포 함수 는 non decreasing 함수이다. 이에 따라, 임의의 균등 확률 변수 에 대해서 의 값을 구하면 수열이 확률 밀도 함수 를 따르는 난수 수열이 된다 그러나, 우리가 사용한 확률함수의 경우, 식으로 표현되지 않을 뿐만 아니라, 연속적인 함수가 아니기 때문에, 누적분포함수의 역함수를 구하는 것이 어렵다. 이를 해결하기 위하여, 를 연속함수로 근사하여 비선형 방정식의 풀이법인 Bisection을 이용해 의 근을 구해 를 얻었다  
이제 정상적으로 작동하는지 여부를 확인하기 위하여, 프로그램으로 구한 난수배열이 확률 밀도 함수 를 따르는지 확인하면 된다.  
문제에서 사용하는 확률 밀도 함수의 경우 정의역이 [0,1]인 함수이므로, 히스토그램의 부분구간을 1/8(0.0075125)로 하여 128개의 구간으로 나누었으며, 난수의 개수는 10,000개로 하였다.  
과제에서 사용할 확률 밀도함수는 아래와 같이 표현이 되며, 총 500개의 값을 샘플링 하였다. 샘플링의 개수가 적을경우, 연속적이지 않은 함수의 특성상, 0의 값을 가지는 구간이 생겨, 샘플링 값을 적당히 크게 만들었다.

<Curve sampling 500개>

  
(프로그램 2-2(a) 이용)을 이용하여 위의 확률그래프의 난수수열에 대한 히스토 그램을 histogram.txt에 저장하였고 저장하였고 (프로그램 2-3 이용) 그 결과는 다음과 같다

<(프로그램 2-2(a) 이용)한 히스토그램>

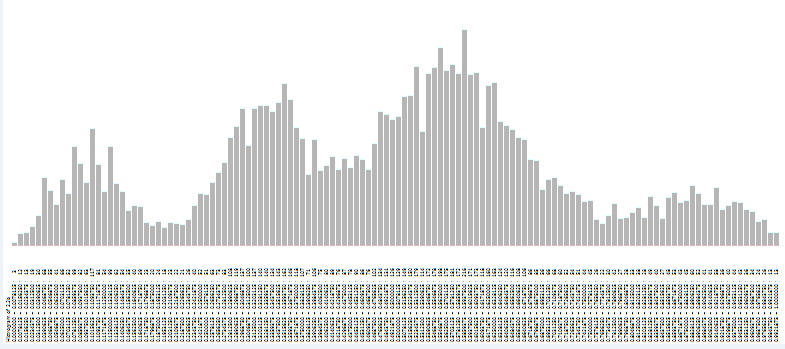


그 그림을 보았을 때, 모양의 형이 서로 비슷한 모양을 따르는 것을 확인 할 수 있다.

1. ***비선형 방정식의 근을 구하는 방법 구현 시 앞에서 사용한 Bisection 방법을 secant방법으로 대치한 난수 생성 프로그램을 작성하라 (프로그램 2-2(b)). 물론 자신의 프로그램이 제대로 작동하는지 실험적으로 확인하라.***

2번과 동일한 방식으로 10000개의 난수수열을 생성하여 히스토그램으로 표현을 해보았다.

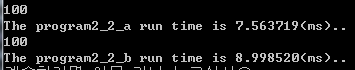
<(프로그램 2-2(b) 이용)한 히스토그램>



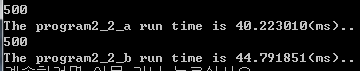
위 그림과 <(프로그램 2-2(a) 이용)한 히스토그램>을 비교하면, 몇몇 부분에서 값이 튀는 양상이 나타나지만, 어렴풋이 비슷한 그래프 형태를 나타내는 것을 볼 수 있다.  
이 방식의 경우, bisection으로 초기 두개의 점을 구한 후, secant 방식을 써서 나머지 값을 구하는 방식이다. Secant 방식의 경우 초기값을 어떻게 정하는지에 따라서, 난수생성 시간에 차이가 존재를 하였는데, 초기값을 정해준 후, 값을 구하는 것과, bisection 을 이용하여 초기값을 구하여 문제의 난수수열을 생성 하는데는 많은 시간의 차이가 나타났다. Bisection 을 이용하여 초기값을 구할 경우, bisection 을 통한 시간과, secant 시간이 포함되어 초기값을 설정한 후 secant 방식으로 값을 구하는 것보다 시간이 많이 걸림을 알 수 있었다. 그럼에도 bisection 을 사용하여 값을 구하게 하는 이유가 무엇인지 알아볼 필요가 있었다. 만약 임의로 초기값을 설정하게 할 경우, 초기값이 정의역의 범위인 [0,1]을 벗어나는 경우가 발생할 수 있기때문에, 이 것을 방지해주는 기능이 들어있었다.

1. ***충분히 큰 난수의 개수 에 대해 프로그램 2-2(a)와 프로그램 2-2(b)를 수행시킨 후, 각 방법이 난수를 생성하는데 걸린 시간을 비교하라.***

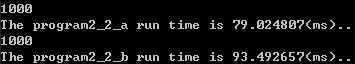
<난수의 개수 : 100개>



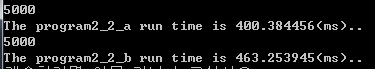
<난수의 개수 :500개>



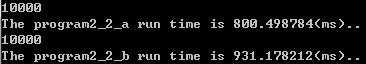
<난수의 개수 :1000개>



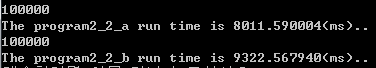
<난수의 개수 :5000개>



<난수의 개수 :10,000개>



<난수의 개수 :100,000개>



모든 경우, bisection을 사용하는 것보다 secant를 사용한게 더 오랜 시간이 걸린 것을 확인할 수 있다.