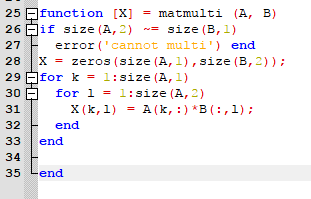
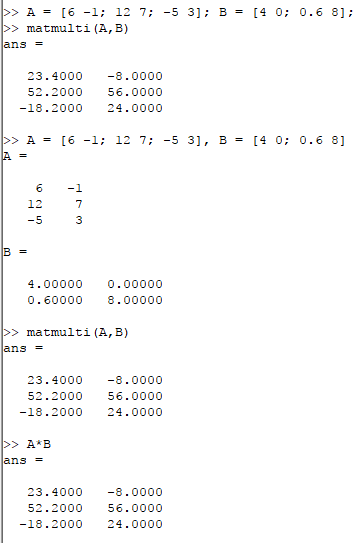
과제 4 201300995 이상건

문제 8.6

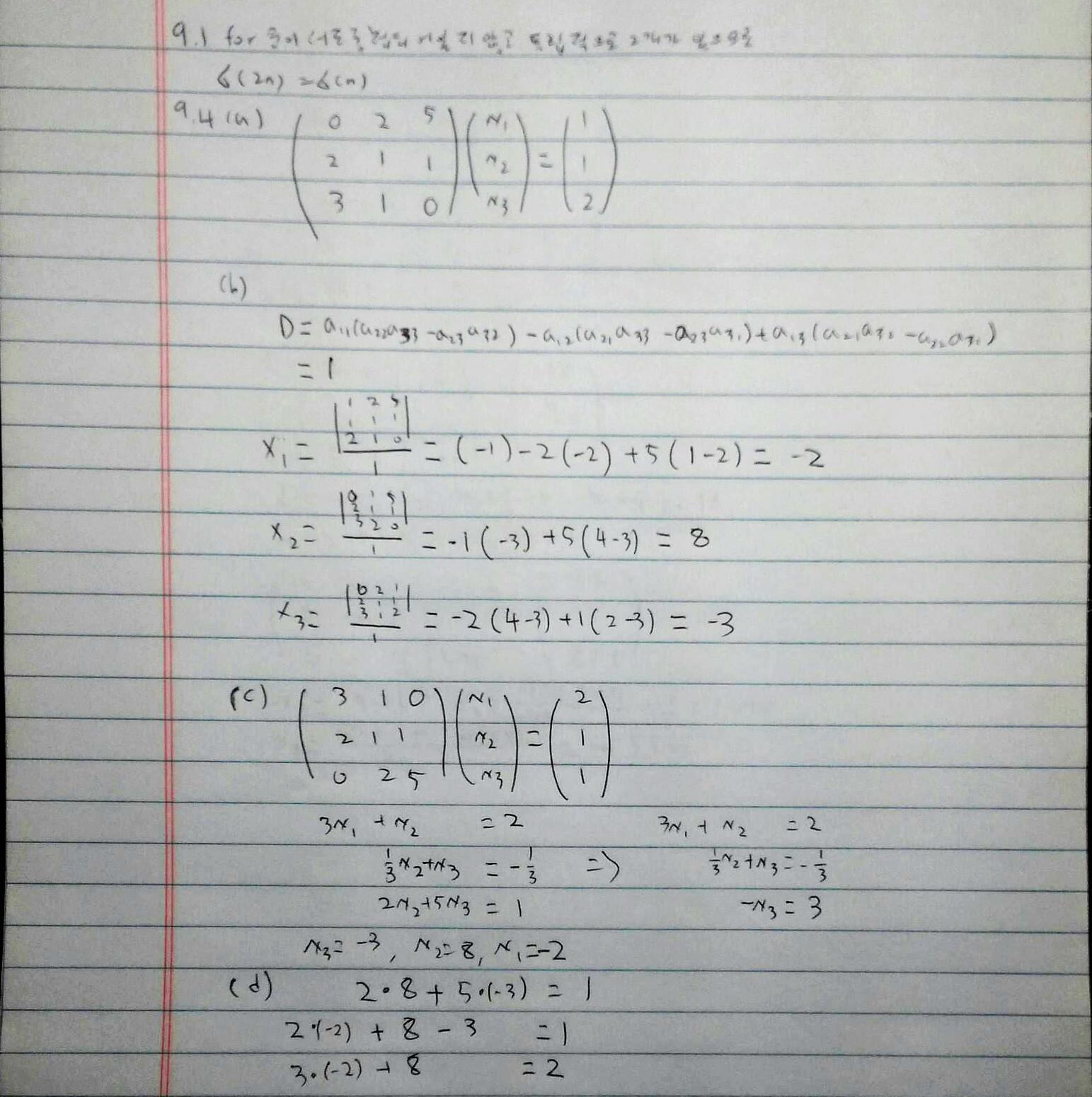


행렬의 곱셈을 구하는 코드 matmulti.m 을 만들었다. 두 행 A,B를 곱할 때 A의 열 개수와 B의 행 개수가 맞는지 확인하고 아닐 시 에러 코드를 내보낸다. 그 뒤 일반적으로 정의되어 있는 행렬의 곱을 수행한다.

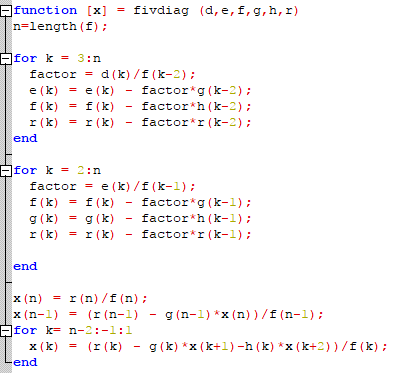


코드 실행결과는 다음과 같다. 실제 행렬 곱과 잘 일치한다.

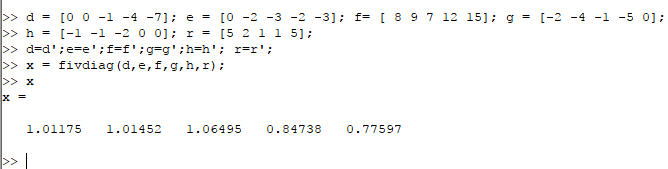
문제 9.1, 9.4



문제 9.16

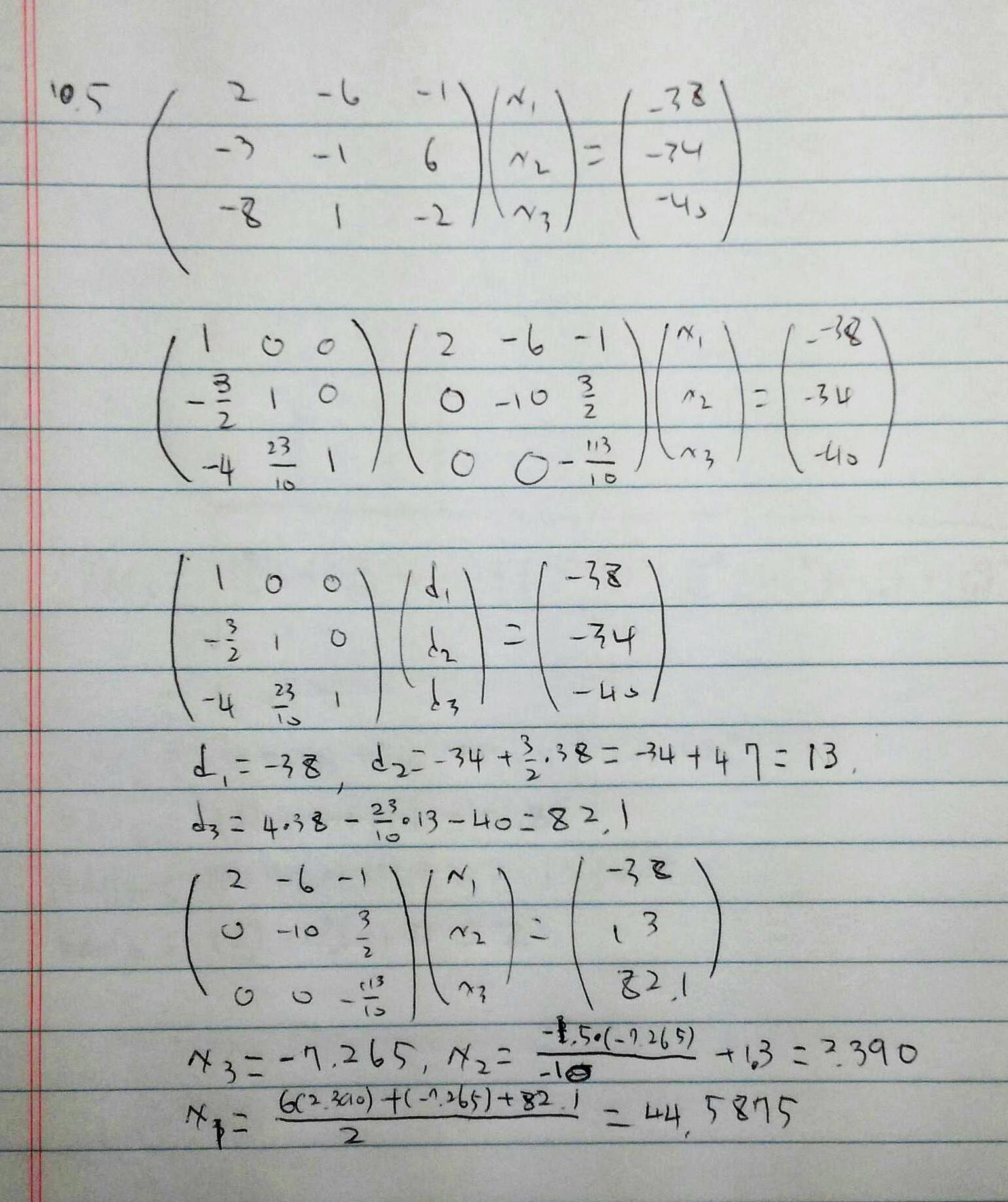


책에 나와있는 Tridiag.m 을 기반으로 하여 fivdiag.m을 작성하였다. 코드의 핵심은 일단 d행렬을 없애 e,f,g,h의 4대각행렬도 나타낸뒤, 전방소거는 3대각행렬과 똑같이 하면 되고, 후방소거의 코드만 살짝 바꿔주면 되었다.

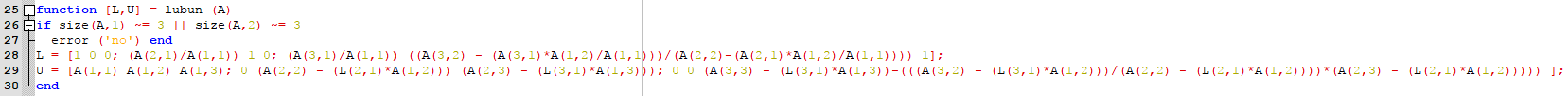


이 코드를 이용하여 구한 값이다. 여기서 이상한 점은 이 해가 1행, 2행, 3행, 4행에서는 정해이지만 5행에선 안된다는 것이다. 코드를 아무리 봐도 잘못된 곳이 없는 것 같은데 이유를 모르겠다.

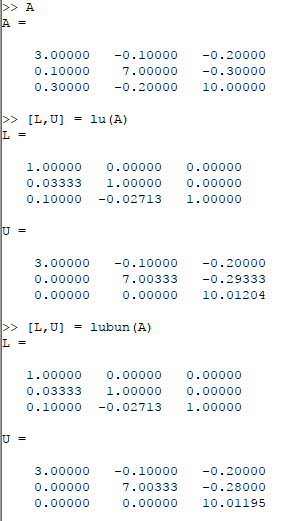
문제 10.5



문제 10.6

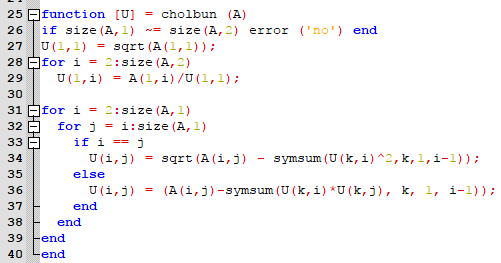


다음과 같이 lubun.m을 만들었다. 특별하게 계산하는 것 없이 LU분해법을 했을 때 L에 들어가는 값과 U에 들어가는 값을 손으로 계산 후 써 넣은 것 뿐이다.



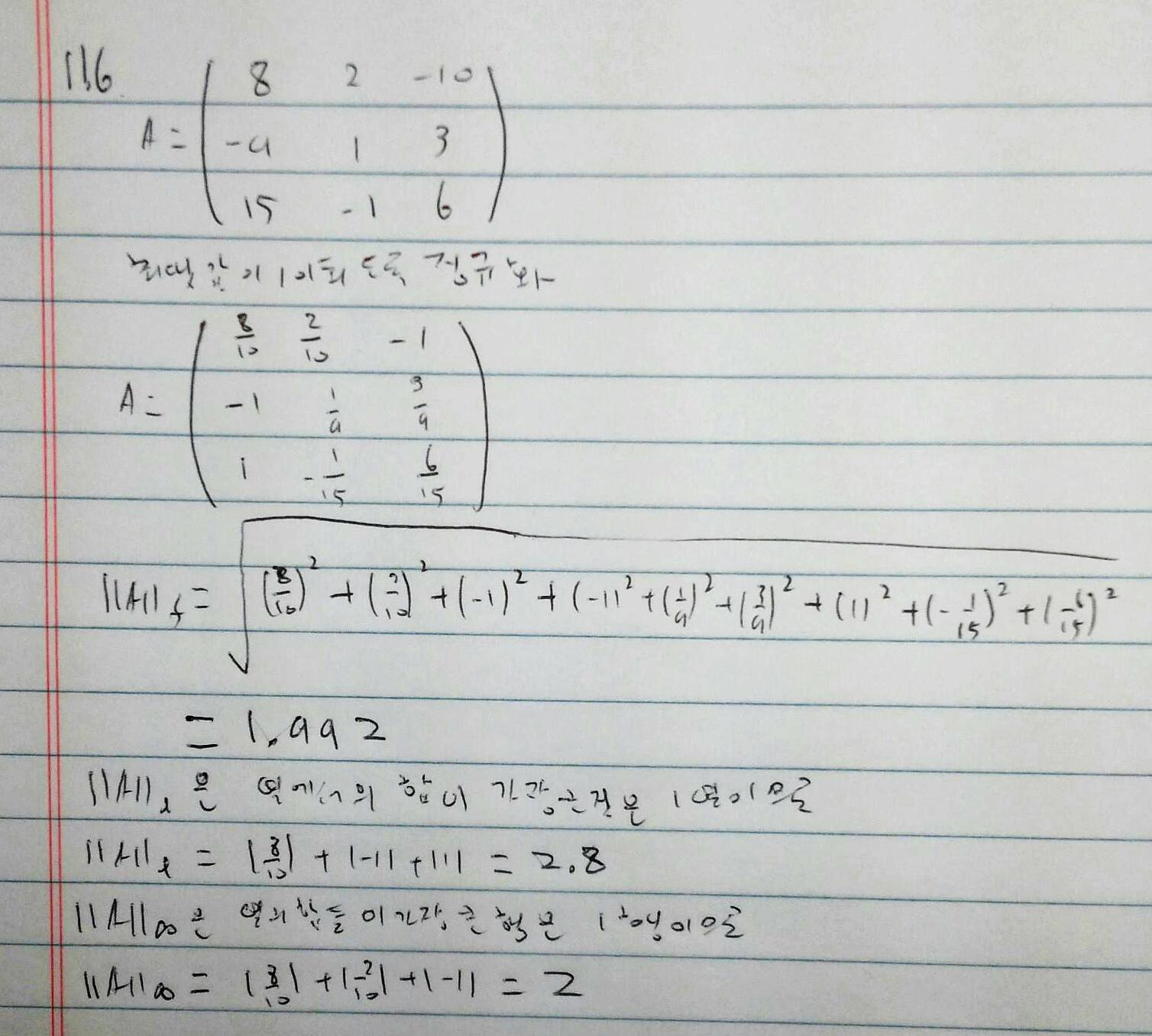
이를 시험했을 때 octave에 내장되어있는 lu함수 결과와 일치함을 알 수 있다. 약간의 오차는 프로그램 상 오류이고 무시할 수 있을 만큼 작다.

문제 10.9

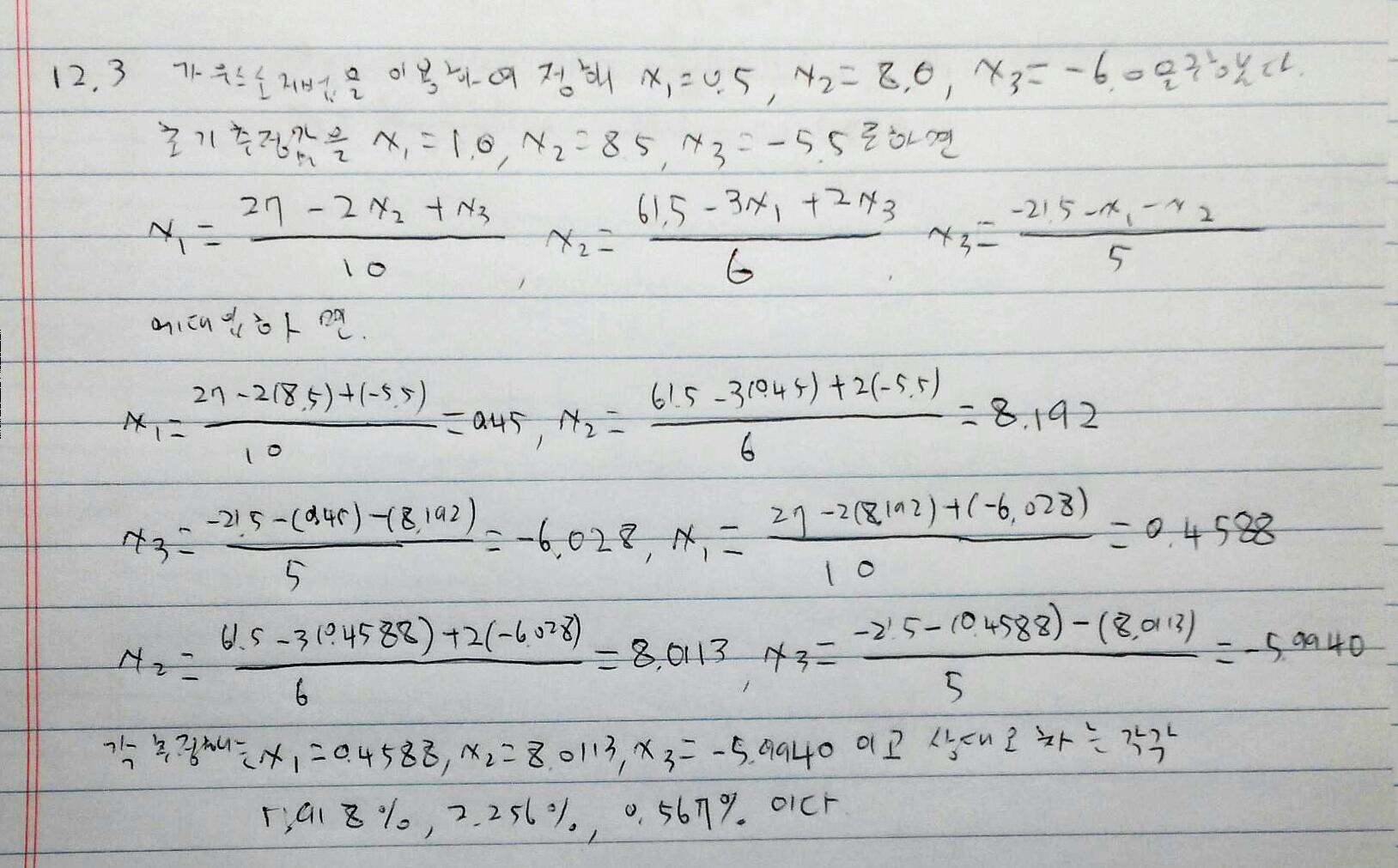


cholbun.m 코드를 작성하였다. Symsum은 시그마 덧셈을 나타내는 코드이다. 내가 작성한 프로그램은 octave환경이므로 symsum을 지원하지 않아 결과를 확인 할 수 없었다. 하지만 수학적으로는 맞으니 matlab에서는 잘 작동될 것이다.

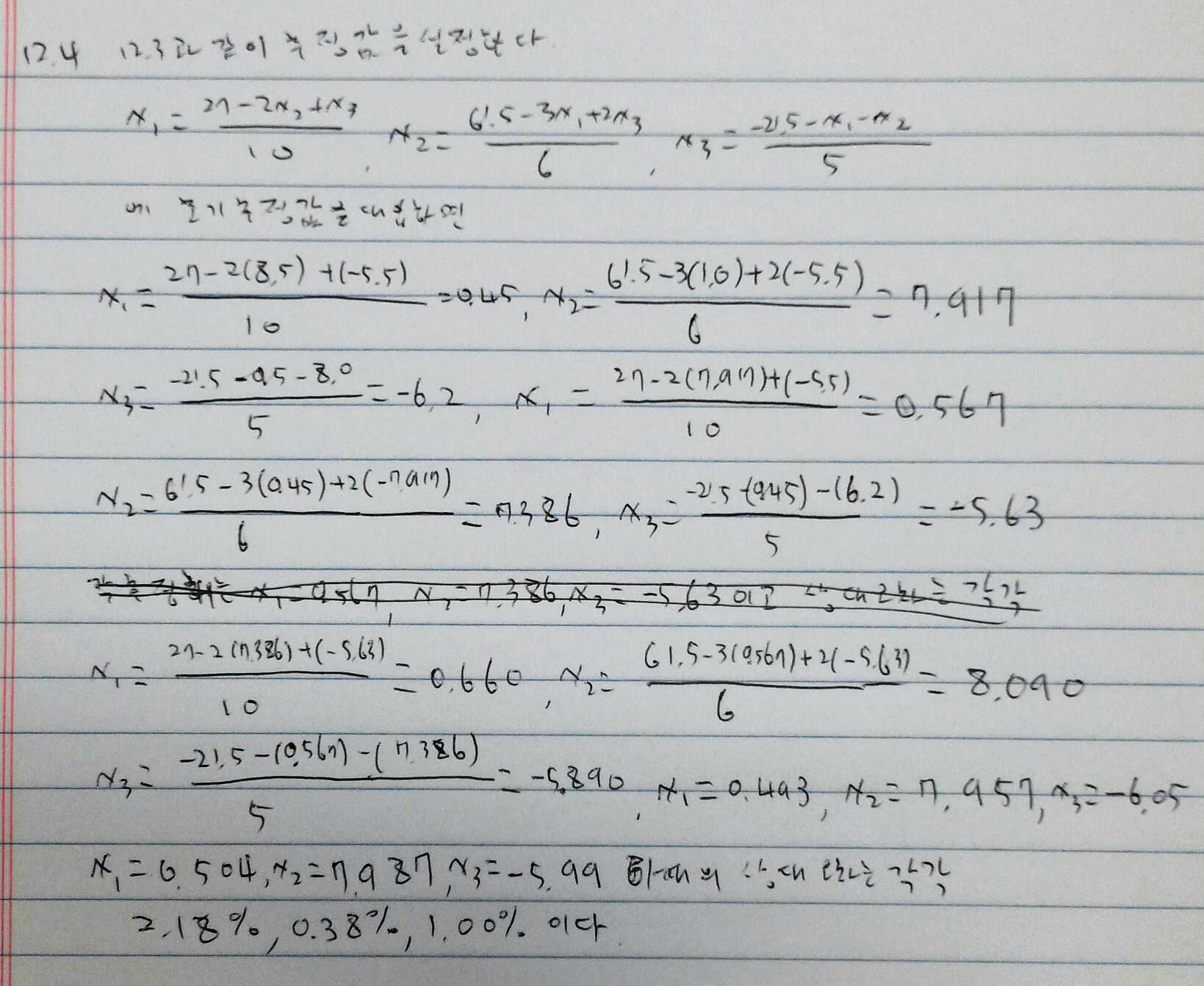
문제 11.6



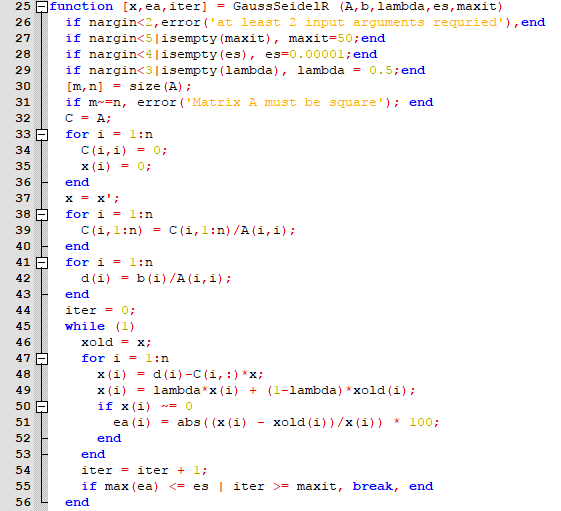
문제 12.3



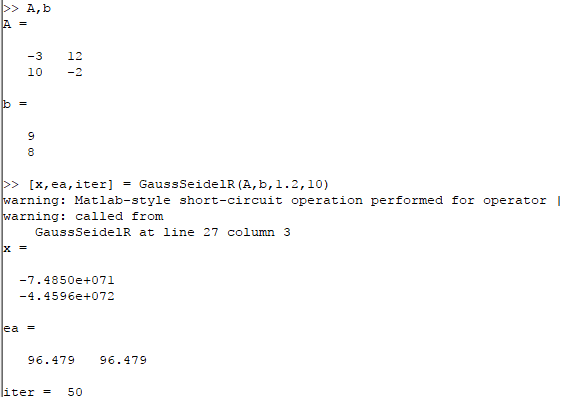
문제 12.4

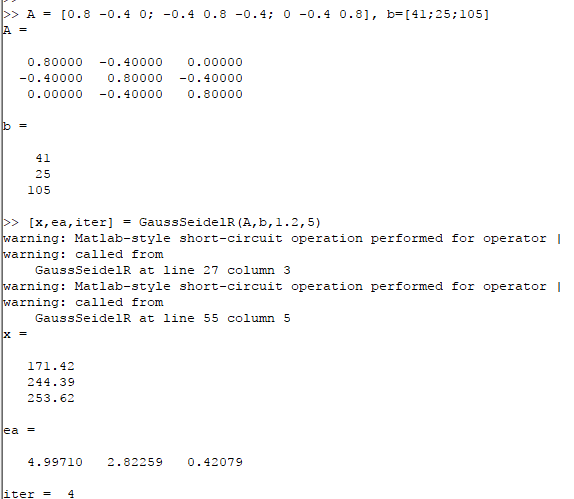


문제 12.13



코드에 특별한 점은 없고 교재에 있는 GaussSeidel.m코드에 x(i) = lambda\*x(i) + (1-lambda)\*xold(i); 한 줄 추가했을 뿐이다. 이것을 예제 12.2와 연습문제 12.2b에 적용시켜 보면





가 된다.