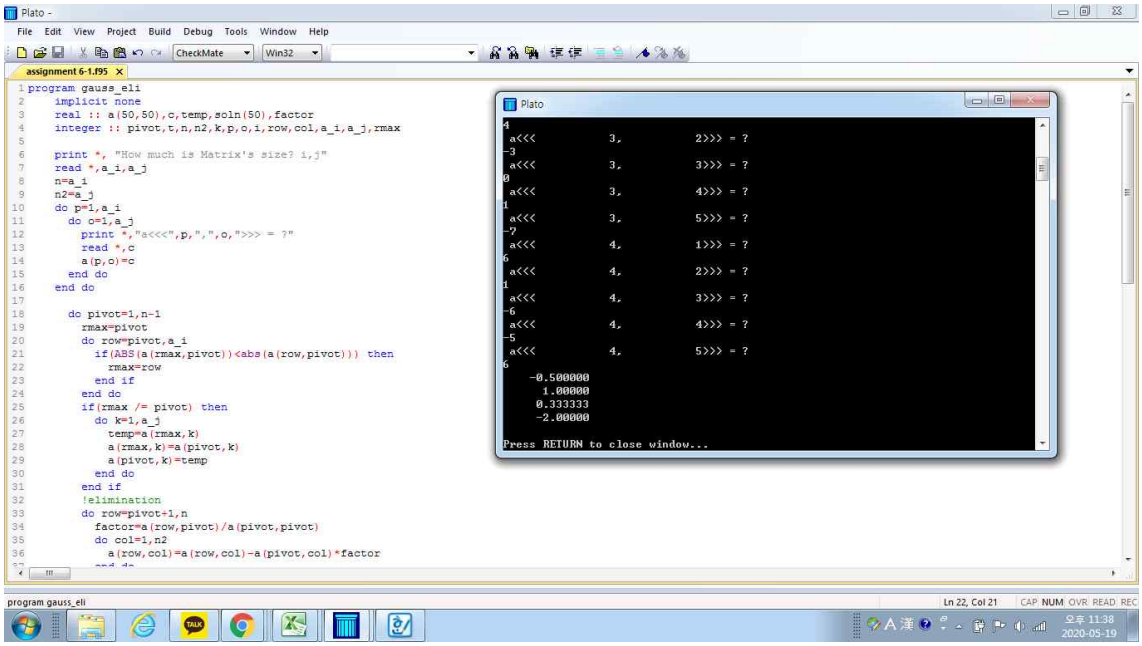
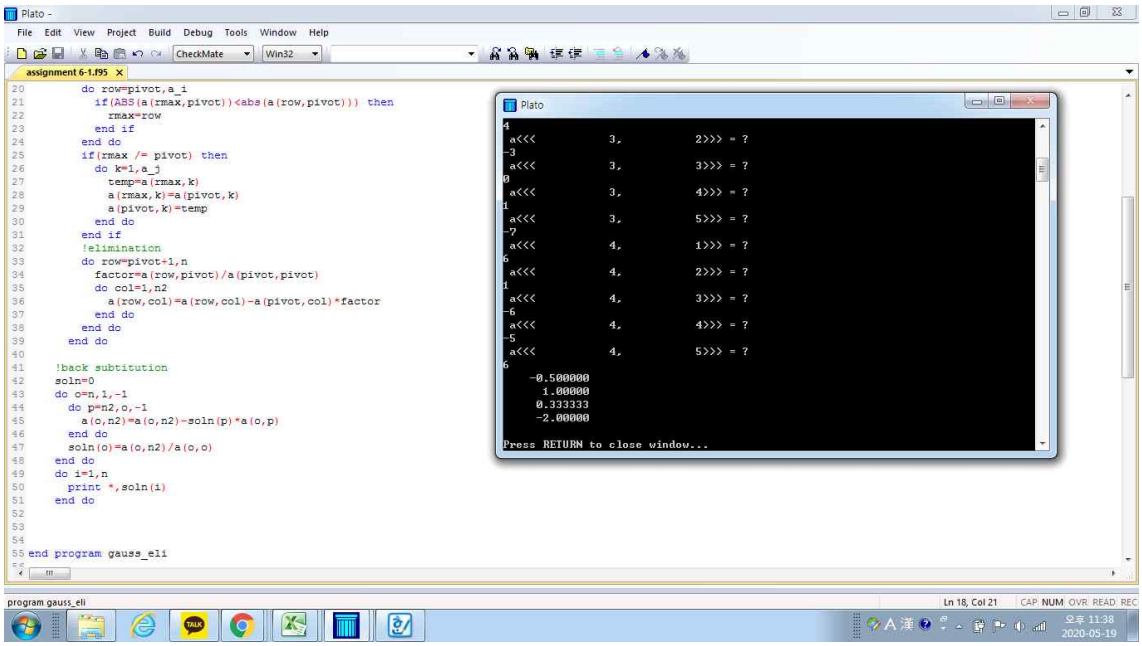
수치 해석 과제#6

2015170305 김태희

과제6-1

1)코드와 결과





2)코드 설명

(1)program gauss_eli를 시작한다.

(2)implicit none을 하고 실수변수로 a(50,50),c,temp,soln(50),factor 변수를 선언하고 정수변수로 pivot,t,n,n2,k,p,o,i,row,col,a_i,a_j,rmax을 선언한다. a는 2차원배열로 행렬 a를 나타내며 soln은 방정식의 해의 값을 1차원 배열로 저장한다. temp는 변수 값 교환 시 사용되며 factor는 pivot,row,col는 이름 그대로를 의미하고 factor는 elimination시 pivot에 곱하

는 값, a_i,a_j는 각각 augmented 행렬의 행과, 열의 개수 n,n2는 각각 이를 저장하는 변수이다. T,k,o,p,i는 do로 반복문을 실행하기 위해 쓰이는 변수이고 rmax는 어떤 열내에서 최대 값을 가지는 row를 의미한다.
(3)read *,a_i,a_j로 augmented matrix의 정보를 받고 그것을 각각 n,n2에 저장한다.
(4)행렬의 성분을 a 변수에 받기위해 do문을 먼저 행의 범위로 p=1,a_i 열의 범위로 o=1,a_j로 하고 print 문의 문자열이 매번 출력되며 그 문자열에 해당되는 행렬의 성분을 입력하면 c로 저장되고 이 값을 a(p,o)로 할당한다.
(5)실제로 elimination이 구현되는 부분은 수업자료에서는 4단 loop문으로 되어있지만 사실 T=1일 때만 해도 elimination이 모두 구현되므로 최외각 do T=1, a_i는 제거하고 do pivot=T, n-1을 do pivot=1, n-1로 수정하였다. pivot은 행렬의 column을 의미하고 이 loop 문은 elimination을 1~n-1 column 까지 진행 할 것이라는 의미이다. rmax는 일단 pivot column과 같게 초기화하고

```
do row=pivot,a_i
          if(ABS(a(rmax,pivot))<abs(a(row,pivot))) then
            rmax=row
          end if
        end do
```

코드를 통해 어떤 col 내에 성분중 절대값이 최대일때의 row값을 rmax로 반환한다.

```
if(rmax /= pivot) then
          do k=1,a_j
            temp=a(rmax,k)
            a(rmax,k)=a(pivot,k)
            a(pivot,k)=temp
          end do
        end if
```

그리고 그 값을 현재 pivot 값을 row값으로 가지는 부분과 교환한다. 이를 rmax 행과 pivot 행에 있는 모든 성분에 대해 실시한다.

```
do row=pivot+1,n
          factor=a(row,pivot)/a(pivot,pivot)
          do col=1,n2
            a(row,col)=a(row,col)-a(pivot,col)*factor
          end do
        end do
```

pivot값에 해당하는 column의 pivot 열에 절대 값이 최대인 성분이 옮겨졌으면 elimination을 진행한다 elimination은 각 row의 값을 pivot에 있는 값으로 나누어 factor를 구한 후 그 factor를 pivot 행에 모든 성분에 곱하여 pivot 행 밑에 있는 모든 행의 모든 성분에 빼서 구할 수 있다. 계산후에 pivot 값의 column의 pivot 값 밑은 모두 '0'으로 소거 될 것이다.

위와 같은 과정을
do pivot=1,n-1
를 통해 column 1부터 n-1까지 시행하면 대각 성분밑으로 모두 0인 행렬이 얻어진다.
(6)soln=0

```
do o=n,1,-1
  do p=n2,o,-1
    a(o,n2)=a(o,n2)-soln(p)*a(o,p)
  end do
  soln(o)=a(o,n2)/a(o,o)
end do
```

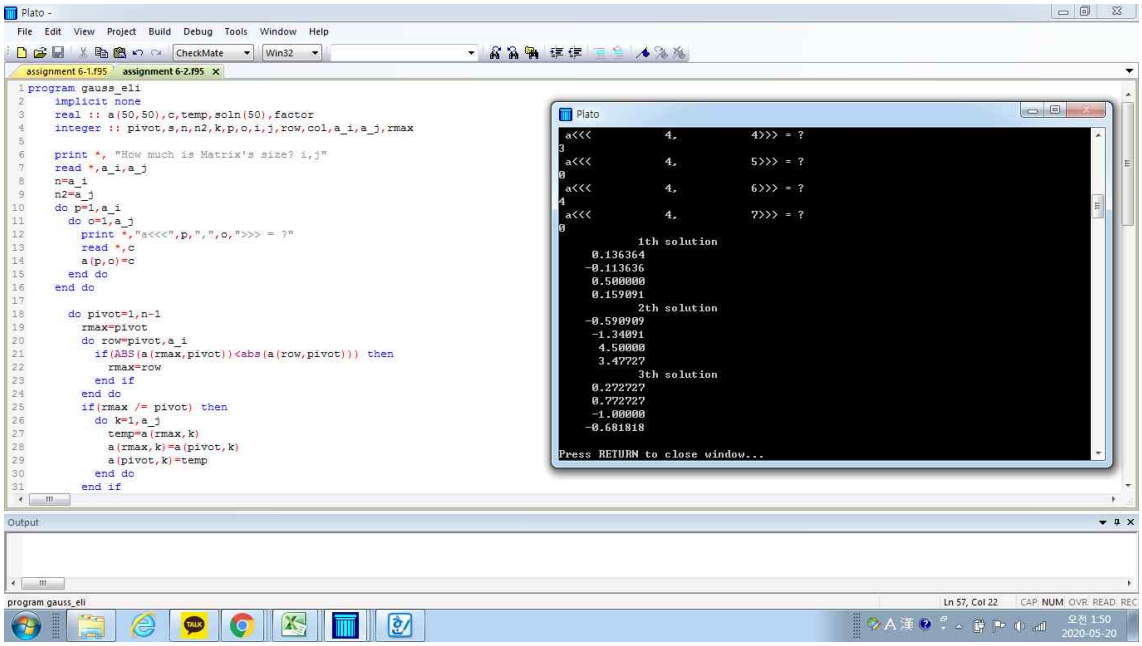
soln행렬을 모두 0으로 초기화 한 후 Ax=b 방정식의 해인 x값을 soln에 저장할 것이다. 먼저 back subtitution을 할 것이며 여기서 o값은 행을 의미하고 p값은 그 행에서 n2열부터 o열까지 역으로 loop문을 쓸 것을 의미한다. a(o,n2)는 elimination 후 우변값을 의미하며 행렬 A값의 대각성분 윗부분을 모두 0으로 만들도록 하면 b의 성분값은 자연스럽게 soln 성분에에 행 o에 pivot을 제외한 성분을 곱하여 빼준것과 같다. 이 과정을 지나면 행렬은 대각성분만이 남게 되고 각각의 대각성분 a(o,o)로 우변성분이 a(o,n2)를 나누어주면 soln(o)즉 해가 나오게 된다 이값을 soln에 입력한다.이
(7)do i=1,n

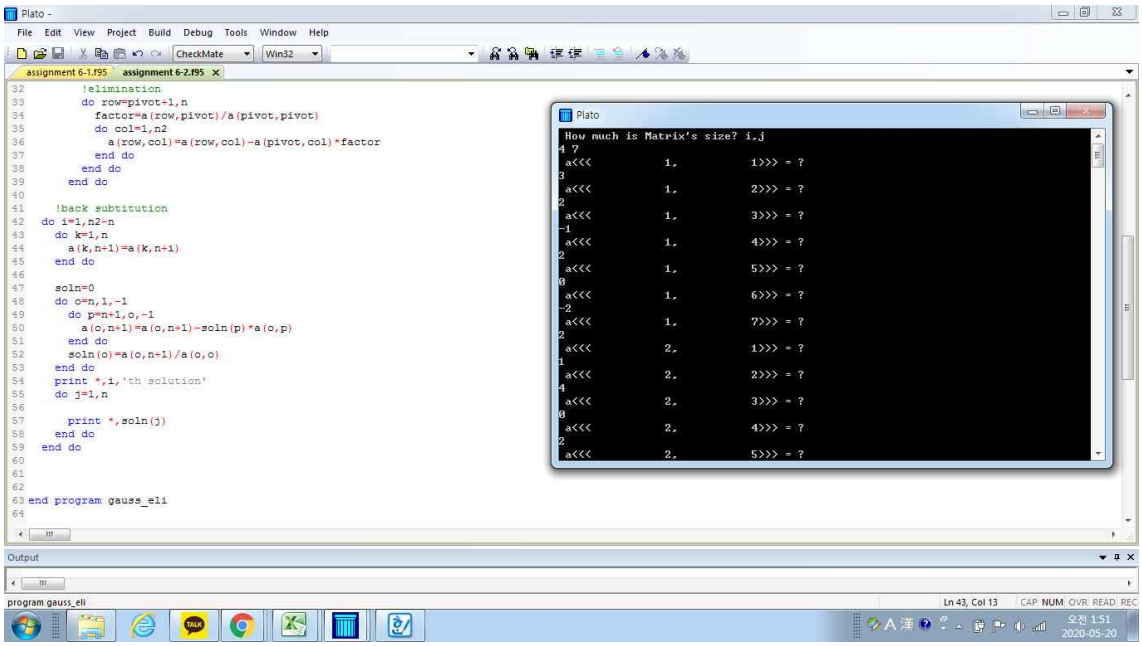
```
  print *,soln(i)
end do
```

모든 해를 출력한다. n값이 행의 수와 일치하므로 모든 해가 출력된다.
(3)수 계산과 비교
수 계산은 뒤쪽에 첨부 하였고 값은 프로그래밍으로 계산한것과 동일하게 나온다.

과제6-2

1)코드와 실행 결과





(2)변형된 코드 부분

먼저 대각 성분 아랫부분의 elimination 은 augmented matrix를 4*7로 만들고 그대로 코드를 진행하면 elimination 이 진행된다.

바뀐 부분은 이후에 back subtitution 부분 이다.

```
do i=1,n2-n
```
먼저 구하려는 경우가 n2-n개이므로 do I=1,n2-n까지 범위로 한다.
```
    do k=1,n
      a(k,n+1)=a(k,n+i)
    end do
```
Ax=b에서 b값이 n+i 열에 있으므로 그 열에 있는 성분을 3번의 시행마다 각각 n+1열로 옮겨와야한다.
```
    soln=0
    do o=n,1,-1
      do p=n+1,o,-1
        a(o,n+1)=a(o,n+1)-soln(p)*a(o,p)
      end do
      soln(o)=a(o,n+1)/a(o,o)
    end do
```
soln=0으로 초기화 한 후 o 변수는 행을 역으로 지정하고 p변수는 n+1열부터 1열까지 열을 역으로 지정한다. 앞에 과정에서 n+1열에 우변의 값을 모두 옮겨왔으므로 이전에 a(o,n2)를 a(o,n+1)로만 바꾸어주면 나머지 과정은 동일하게 진행된다.
```
    print *,i,'th solution'
    do j=1,n

      print *,soln(j)
    end do
  end do
```
구하려는 방정식의 해는 각각 print *, soln(j)로 출력되며 이 경우에는 soln이 4개씩 나올것이며 서로 다른 3개의 방정식에서 각각 진행되어 출력될 것이다.

**수계산 부분