7.4 The Dimension Theorem and Its Implications

Exercise 7.5. (*Rank and Nullity*)

(a) Use the MATLAB command *rank* and the Formula (2) in Section 7.4 to find the nullity of the matrix

$$A = \begin{bmatrix} 3 & 2 & 1 & 3 & 5 \\ 6 & 4 & 3 & 5 & 7 \\ 9 & 6 & 5 & 7 & 9 \\ 3 & 2 & 0 & 4 & 8 \end{bmatrix}.$$

(b) Confirm that the result obtained in (a) is consistent with the number of basis vectors which are obtained by using the MATLAB command *null*.

Solution.

(a)
```
% Set A.
A = [3 2 1 3 5; 6 4 3 5 7; 9 6 5 7 9; 3 2 0 4 8];

% Find the rank of A by using the command rank.
rank_A = rank(A);

% Size of the matrix A.
[ m n ] = size(A);
% m = the number of rows of A, n = the number of columns of A.

 % By (2) in section 7.4, rankA + nullA = n.
null_A = n - rank_A;

disp('The nullity of A is'); disp(null_A);
```

MATLAB results.

```
The nullity of A is
     3
```

(b)
```
% Set A.
A = [3 2 1 3 5; 6 4 3 5 7; 9 6 5 7 9; 3 2 0 4 8];

% Find a basis for the null space of A.
nullA = null(A,'r');
% null(A,'r') returns a matrix
% whose columns are a basis for the null space of A.

[m n] = size(nullA);
% Since the number of columns of nullA is n,
```

```
% thus, n = the number of basis vectors of the null space of A.

disp('The nullity of A is'); disp(n);
```

MATLAB results.

```
The nullity of A is
     3
```

7.5 The Rank Theorem and Its Implications

Exercise 7.6. Note that the rank of a nonzero matrix A is equal to the order of the largest square submatrix of A (formed by deleting rows and columns of A) whose determinant is nonzero. In this problem, we make a function file `CheckRank.m` to find the rank of the given matrix using this fact. We want to obtain the execution results as follows:

```
>> A=[1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16];
>> rankA=CheckRank(A)
rankA =
     2
```

For this, you may start with the largest square matrices to be found in A and a search is started for the first submatrix with a nonzero determinant. Use the MATLAB command *nchoosek* to select all the combinations of rows and columns needed in the search process and you may use the several MATLAB commands if you need. Complete the m-file below and check the determinant of the matrices A, B, and C given in the Exercise 7.3 (b). Also, compare the results using the MATLAB command *rank*.

Solution.

```
%--- function file 'CheckRank.m' ---%
function [rank_A]= CheckRank(A)

    [m,n]=size(A);  % size of given matrix
    flg=1;  % flag for while loop
    if m>n  % if (# of row) > (# of col)
        A=A';
    end
    A=sym(A);   % Set A as a symbolic object

    K = min(m,n);   N = max(m,n);   % k : row number, N: col number
    k=K;    % from the largest size of submatrix
    while flg == 1
        comb_row=nchoosek(1:K, k);  % combinations of row
        comb_col=nchoosek(1:N, k);  % combinations of columns
        for ii=1:size(comb_row) %
            selected_A=A(comb_row(ii,:),:); % selected row index
            for jj=1:size(comb_col)
```

```
                sub_A=selected_A(:,comb_col(jj, :));   % selected col index
                if det(sub_A)~=0    % if non zeros determinant appears
                    rank_A=k;   % the size at that time <- rank
                    flg=0;  % stop the while loop.
                end
            end
        end
        k=k-1;  % if all submatrices of size k have a zero determinant,
                % reduce the size of submatrix.
    end


end
```

To check the determinant of the matrices A, B, and C given in the Exercise 7.3, you execute the followings:

```
A=[3 2 1 3 5; 6 4 3 5 7; 9 6 5 7 9; 3 2 0 4 8];
B=[3 -1 3 2 5; 5 -3 2 3 4; 1 -3 -5 0 -7; 7 -5 1 4 1];
C=[1 3 2 1; -2 -6 0 -6 ;3 9 1 8; -1 -3 -3 -6; 1 3 2 1; 4 12 1 11];

fprintf('my rank(A): %.5f, MATLAB rank(A): %.5f \n', CheckRank(A), rank(A));
fprintf('my rank(B): %.5f, MATLAB rank(B): %.5f \n', CheckRank(B), rank(B));
fprintf('my rank(C): %.5f, MATLAB rank(C): %.5f \n', CheckRank(C), rank(C));
```

MATLAB results.

```
my rank(A): 2.00000, MATLAB rank(A): 2.00000
my rank(B): 3.00000, MATLAB rank(B): 3.00000
my rank(C): 3.00000, MATLAB rank(C): 3.00000
```

Those are the same results as given in Exercise7.3.