Math 236 **Problem Set 7** **Spring, 2001**

Due date: Wednesday, April 4 but #9 due Friday, April 6

1. In each part, if the vectors are not linearly independent, then *also* (i) find a nontrivial relation among them, *and* (ii) express some one of them as a linear combination of the others.

   (a) Do page 119, Exercise 15.

   (b) Do page 119, Exercise 16.

2. (a) Do page 119, Exercise 23. Use the *definition* of subspace here!

   (b) Do page 119, Exercise 24.

3. (a) Let A be the matrix of page 131, Exercise 22. Find bases of the null space of A and the column space of A and determine the dimensions of $N(A)$ and $C(A)$.

   (b) Let $T\colon \mathbb{R}^5 \to \mathbb{R}^4$ be defined by

$$T\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}\right) = \begin{bmatrix} x_1 + 2x_2 + x_3 + 2x_4 + x_5 \\ x_1 + 2x_2 + 2x_3 + x_4 + 2x_5 \\ 2x_1 + 4x_2 + 3x_3 + 3x_4 + 3x_5 \\ x_3 - x_4 - x_5 \end{bmatrix}.$$

   Find bases of $\ker(T)$ and $\operatorname{im}(T)$ and determine the dimensions of $\ker(T)$ and $\operatorname{im}(T)$.

4. (a) Find a basis of the subspace V of $\mathbb{R}^4$ that is spanned by

$$\begin{bmatrix} 1 \\ 1 \\ 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \\ 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 3 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \\ 3 \\ -1 \end{bmatrix}.$$

   (b) Do the given vectors in (a) form a basis of $\mathbb{R}^4$? Why or why not?

5. In each part, also describe as simply as possible—geometrically or otherwise—the subspace of $\mathbb{R}^3$ involved.

   (a) Do page 130, Exercise 6.

   (b) Do page 131, Exercise 16.

6. Do the vectors $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \\ 3 \end{bmatrix}$ form a basis of $\mathbb{R}^3$? Why or why not?

7. Do page 131, Exercise 30.

8. Do page 132, Exercise 38 (b).

9. *(Counts as two problems!)* Define and test sufficiently a Mathematica function `columnSpace` that finds a basis of the column space of any matrix.

   If `mat` is any matrix (not necessarily square), then `columnSpace[mat]` returns as result the basis of the column space of the matrix consisting of its "pivot columns". This basis should be returned as a list of these columns as ordinary lists—*not* as the columns of a matrix. For example:

```
   A = { {1,2,0,3,4,1}, {3,6,1,17,18,0}, {4,8,0,12,16,2}, {3,6,0,9,12,1} };
   A // MatrixForm
1 2 0  3  4 1
3 6 1 17 18 0
4 8 0 12 16 2
3 6 0  9 12 1
   columnSpace[A]
{ {1, 3, 4, 3}, {0, 1, 0, 0}, {1, 0, 2, 1} }
   Transpose[ columnSpace[A] ] // MatrixForm
1 0 1
3 1 0
4 0 2
3 0 1
```

Your testing should include enough examples to demonstrate that your function gives the correct result in varied and even "unusual" cases—when all the columns of the original matrix constitute the basis; when the basis is empty; when there are more columns than rows; when there is only one row; when the matrix is already in reduced row-echelon form; when it is not; etc. To demonstrate in each case that your answer is correct, you can use `RowReduce` and manually mark what the pivot columns are, then indicate that your function `columnSpace` gives those same vectors.

Throughout this problem, you should avoid all loops formed with `Do`, `For`, and `While`; you should also avoid all conditional constructions involving `If` (and `Which` and `Switch`). One purpose of this problem is to get you programming in an array-oriented and "functional" style. So look for places where you can use MATHEMATICA's `Apply` and `Map`.

Modularize—break up into smaller chunks—your work of defining `columnSpace` by also defining (at least) two auxiliary functions:

- A function `isolateLeading1` used in the form `isolateLeading1[vec]`, where `vec` is a vector of the form you get as a *row* of a reduced row-echelon matrix, that returns as its result the vector obtained by changing to `0` each entry of `vec` after a leading `1`. For example:

```
   isolateLeading1[{0, 0, 1, 2, 0, 1}]
{0, 0, 1, 0, 0, 0}
   isolateLeading1[{0, 0, 0, 0, 0, 1}]
{0, 0, 0, 0, 0, 1}
   isolateLeading1[{0, 0, 0, 0, 0, 0}]
{0, 0, 0, 0, 0, 0}
```

- A function `markPivotColumns` used in the form `markPivotColumns[echmat]`, where `echmat` is a matrix that is already in reduced row-echelon form, that returns as result a "Boolean" vector—a vector consisting entirely of 0's and 1's—whose length is the number of columns of `echmat` and which has a 1 in the position of each pivot column of `echmat` and a 0 in every other position. For example:

```
   reduced = { 1,2,0,3,4,0}, {0,0,1,8,6,0}, {0,0,0,0,0,1}, {0,0,0,0,0,0} };
   reduced // MatrixForm
1 2 0 3 4 0
0 0 1 8 6 0
0 0 0 0 0 1
0 0 0 0 0 0
   markPivotColumns[reduced]
{1,0,1,0,0,1}
```

Here's a suggested way to define `markPivotColumns`. First use `isolateLeading1` on each column of the argument (a reduced row-echelon matrix); there's a way to do that "all at once", without using any loop. Now how can you tell which columns of what you obtained are in fact pivot columns? Look at the preceding example. After using `isolateLeading1` on each of its rows, you would have:

```
1 0 0 0 0 0
0 0 1 0 0 0
0 0 0 0 0 1
0 0 0 0 0 0
```

From that, how can you obtain the following Boolean vector you want?

```
{1,0,1,0,0,1}
```

(You are going to use `markPivotColumns` to tell which columns of a given matrix are its pivot columns. Of course, you first have to use `RowReduce` with a given matrix before you can use `markPivotColumns`.)

In your definition of `columnSpace`, you should form the reduced row-echelon form of the argument matrix and use `markPivotColumns` to indicate which columns of the *original* argument matrix to select. So all you have to do is to select them. For that use my function `booleanSelect`, available in notebook `REcolumnSpace.nb`.

My function `booleanSelect` is used in the form `booleanSelect`[v, b], where `v` is any list and `b` is a Boolean vector having the same length as `v`; its result is the list obtained by selecting those elements of `v` that correspond to the 1's in `b`. For example:

```
   booleanSelect[{2,5,3,4,9,6}, {1,0,1,0,0,1}]
{2,3,6}
```

In your actual use of `booleanSelect`, its first argument will be the list consisting of the *columns* of the original matrix—the one for which you want to find a basis of its column space. For example, in the case of the matrix `A` above, you would want:

```
   booleanSelect[ Transpose[A], {1,0,1,0,0,1} ]
{{1,3,4,3}, {0,1,0,0}, {1,0,2,1}}
```

This last result is the desired list of pivot columns of `A`.