The Complete Solution To $A\mathbf{x} = \mathbf{b}$ (Part I)

Tommy Luckner
Department of Mathematics

Overview

This is a project, extending over two lab meetings, in which we will implement a sophisticated algorithm for finding the complete solution to $A\mathbf{x} = \mathbf{b}$ (if it has a solution). Comprehensive usage of the `for` and `if` structures are required. Several new MATLAB commands are introduced. One additional goal is to convert pseudocode into formal MATLAB code.

Activities

- Shortening vectors:

```
>> v = [-7 -2 -6 0 4]
>> v(1) = ''    % delete the 1st element of v
>> u = [3 3 -5 -3 -7]
>> u([1 2 5]) = ''    % delete 1st, 2nd, and 5th elements of u
```

- Extending vectors:

```
>> v = []    % create a vacant vector
>> v = [v,3]    % concatenate a vector
>> v = [v,[9 -4]]
```

- **break;** terminates the execution of a `for` or `while` loop. Statements in the loop that appear after the `break` statement are not executed. In nested loops, `break` exits only from the loop in which it occurs. Control passes to the statement that follows the end of that loop.

- **fprintf(*format string*, *data*);** formats output, where *`format string`* describes the format of the output fields. Below `%d` denotes that we will input a variable of type double in that place. We then specify what that value is with commas after the line we are printing. `\n` will move down to the next line. For example, type

```
>> y = [-1 -3 -6 7];
>> fprintf('The second element in y is %d.\n',y(2));
>>bob_age=23;
>>sally_age=19;
>>fprintf('Bob is %d years old, and Sally is %d years old. \n',
bob_age, sally_age);
```

  Feel free to look up other ways to display things in MATLAB and other features of the display command, but these should suffice for this lab.

If you need to display say the elements in a vector, a for loop can be used. For example:

```
>>u=[1 2 3];
>>fprintf('The elements of u are:');
for i=1:length(u)
fprintf(' %d',u(i));
end
fprintf('\n'); %This line puts the cursor on the next line
```

- Pseudocode is an informal description of an algorithm that uses a combination of a programming language and ordinary language. It is intended to be read by human beings as opposed to machines. The purpose is to easily outline an algorithm without writing it formally.

In-Class Exercise

Goal (for today): For a given matrix A, determine its reduced row echelon form and list its rank and pivot and free variables.

Algorithm: Create vectors *pivot* and *free* which store the index of the pivot and free variables, respectively. At the end of the algorithm, the rank and pivot and free variables should be printed.

1. Initialize these vectors by assuming that none of the variables are pivot, and all of them are free.
2. Find R, the reduced row echelon form of A. You can use MATLAB's built in function `rref`.
3. Find where the the pivots are located in R. You know how to do this by looking at it. Now turn this procedure into an algorithm.
4. Store the location of these pivots in the vector *pivot*. Make sure to store which index of the pivot in R corresponds to appropriate element in x.
5. Remove all the elements in *pivot* from *free*.
6. Print the results.