ACM/IDS 104 - Problem Set 1 - MATLAB Problems

Before writing your MATLAB code, it is always good practice to get rid of any leftover variables and figures from previous scripts.

```
clc; clear; close all;
```

NOTE: As this is the first problem set (and many of you might be unfamiliar with MATLAB) we will provide some helper code. As the term progresses (and you become more experienced) we will omit this.

Problem 6 (10 points) Solving Linear Systems

We have the matrix:

$$B = \begin{pmatrix} 1 & 2 & \cdots & n \\ n+1 & n+2 & \cdots & 2n \\ \vdots & \vdots & & \vdots \\ n^2-n+1 & n^2-n+2 & \cdots & n^2 \end{pmatrix}$$

Part (a) (5 points)

In this part, your task is to find $\text{rank}(B)$. As mentioned in the problem set, MATLAB is not needed to obtain the answer. However, we can use MATLAB to make a right guess and check our answer. To do this, we first need to construct matrix B in MATLAB:

NOTE: Although you can check your answer here, you still need to justify and show your reasoning to obtain full credit :)

```
n = 100; % set n as specified in Part (b)
B = 1 : n;
for i = 2 : n
    B(i, :) = B(i-1, :) + n;

end
r = rank(B); % check your answer here
```

Part (b) (5 points)

Set $n = 100$ and consider the system of linear equations $Bx = c$ where $c = (1 \quad 2 \quad \cdots \quad n)^T$. Find a solution x such that its first $[n - \text{rank}(B)]$ components are zero. What are the non-zero components of x?

HINT: The backslash operator $B\backslash c$ issues a warning if B is nearly singular and raises an error condition if it detects exact singularity. In that case, use `pinv(B)*c` for finding a particular solution of $Bx = c$. The function `pinv(B)` returns the "pseudoinverse" of B (will discuss the Moore-Penrose pseudoinverse in lecture 16). Also, the following built-in function may be useful: `null`.

```
%{
Let us start by defining the column vector c as specified above.
Remember that in MATLAB we can use ' to transpose a vector.
```

```
%}

c = (1:n)';
%{
Now, we obtain a particular solution, x_0, as described above
%}
x_0 = pinv(B)*c;

%{
Use null(B) to define the matrix V, whose columns form an orthonormal
basis in the vector space of all solution of the homogeneous
system Bx=0.
%}
V = null(B);

%{
Use the rank, r, found in part (a) to define k = n - rank(B).
This is the number of free variables / dimension of the vector
space.
%}
k = n - r;
```

This is a good point to review what we have done so far. Recall that the desired solution of the system is:

$$x = x_0 + V\alpha \quad (\star)$$

where α is a $k \times 1$ vector. We can obtain α by solving the system:

$$x_0 + V\alpha = 0 \quad (\star\star)$$

```
%{
Find alpha by solving the described system (**).
-> Hint1: Remeber that alpha is a k*1 vector. Hence, you need to
restrict the sizes of x_0 and V
-> Hint2: Use backslash
%}
a = V(1:k, :)\-x_0(1:k);
%{
Finally, put everything together and find x using (*)
Use disp(x) to display your solution.
%}
x = x_0 + V*a
```

```
x = 100×1
   -0.0000
    0.0000
   -0.0000
   -0.0000
    0.0000
   -0.0000
    0.0000
    0.0000
   -0.0000
```

```
    -0.0000
      ⋮
      ⋮
```

```
%{
Now, let us see how x compares to the actual solution.
Un-comment the following 2 lines of code once you reach this part.
%}
error = norm(B*x - c);
disp(error);
```

```
    1.0274e-12
```

Don't forget to report the non-zero components of x!