ACM 104 HW #1 Om Sanan

1. Matrix Multiplication

Let $A \in \mathbb{R}^{m \times p}$ and $B \in \mathbb{R}^{p \times n}$

$$A = [a_1 \dots a_p],$$

where each $a_k \in \mathbb{R}^m$.

Now, $\begin{bmatrix} b^1 \\ \vdots \\ b^p \end{bmatrix}$

where each $b^k \in \mathbb{R}^{1 \times n}$ is a row vector.

Define:

$$S = \sum_{k=1}^{p} a_k b^k$$

Each term $a_k b^k$ is an outer product of a column and a row vector, hence an $m \times n$ rank 1 matrix. The i,j th entry of S is:

$$S_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj} = (AB)_{ij}$$

Thus,

$$\boxed{AB = \sum_{k=1}^{p} a_k b^k.}$$

In particular, if $B = x \in \mathbb{R}^p$ is a column vector w/ components $x_1, \dots, x_p$, then each b^k is just the scalar x^k, and

$$Ax = \sum_{k=1}^{p} a_k x_k$$

2. Matrix Identities

a) Guess: $\boxed{A^{-1} - B^{-1} = A^{-1}(B-A)B^{-1}}$

Proof: $A^{-1}(B-A)B^{-1} = A^{-1}BB^{-1} - A^{-1}AB^{-1}$

$= A^{-1}I - IB^{-1}$

$= A^{-1} - B^{-1}$

b) $\frac{d}{dt}\left[A(t)A^{-1}(t)\right] = \frac{d}{dt}[I] = 0$

Applying product rule to LHS:

$$A'(t)A^{-1}(t) + A(t)(A^{-1}(t))' = 0$$

$$\boxed{(A^{-1}(t))' = -A^{-1}A'(t)A^{-1}(t)}$$

c) $A^2(t) = A(t)A(t)$

$$d[A(t)A(t)] = \boxed{A'(t)A(t) + A(t)A'(t) = A^{2}(t)}$$

3. Permuted LU Decomposition

$$A_n = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{bmatrix}$$

Since a_n is symmetric positive definite and of this form: $P_n = I_n$. For $A_n = L_n U_n$, we want:

$$L_n = \begin{bmatrix} 1 & & & 0 \\ l_2 & 1 & & \\ & l_3 & \ddots & \\ 0 & & \ddots & 1 \end{bmatrix}, \quad U_n = \begin{bmatrix} u_1 & -1 & & 0 \\ & u_2 & -1 & \\ & 0 & \ddots & \ddots \\ & & & u_n \end{bmatrix}$$

For row 1: $u_1 = 2$; For row i: $\begin{cases} l_i u_{i-1} = -1 \\ l_i(-1) + u_i = 2 \end{cases}$

Solving, $-\frac{1}{u_{i-1}}(-1) + u_i = 2 \longrightarrow u_i = 2 - \frac{1}{u_{i-1}}$

$u_1 = 2 = \frac{2}{1}$; $u_2 = 2 - \frac{1}{2} = \frac{3}{2}$; $u_3 = 2 - \frac{2}{3} = \frac{4}{3}$; $u_4 = 2 - \frac{3}{4} = \frac{5}{4}$

Pattern: $u_i = \frac{i+1}{i} \longrightarrow l_i = -\frac{1}{u_{i-1}} = \frac{1-i}{i}$

$$L_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -\frac{1}{2} & 1 & 0 & \cdots & 0 \\ 0 & -\frac{2}{3} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad U_n = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ 0 & \frac{3}{2} & -1 & \cdots & 0 \\ 0 & 0 & \frac{4}{3} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{n+1}{n} \end{bmatrix}$$

4. Types of Matrices

a) *Permutation matrix has one 1 in each row and col.

$$(PP^T)_{ij} = \sum_{k=1}^{n} P_{ik}(P^T)_{kj} = \sum_{k=1}^{n} P_{ik} P_{jk}$$

Since row i has one 1 at some position k, and col $j=i$ of P^T also has same 1 at position k:

$$(PP^T)_{ii} = \sum_{k=1}^{n} P_{ik} P_{ik} = 1^2 = 1$$

$\therefore PP^T = I \rightarrow P^T = P^{-1} \rightarrow P$ is orthogonal.

b) No, an orthogonal matrix is not necessarily a permutation matrix.

Counterexample:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H^T H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = I$$

So h is orthogonal. However H has entries $\frac{1}{\sqrt{2}} \neq 0, 1$, so H is not a permutation matrix.

c) Existance: Let $S = \frac{1}{2}(A + A^T)$, $J = \frac{1}{2}(A - A^T)$

Symmetric: $S^T = \frac{1}{2}(A + A^T)^T = \frac{1}{2}(A^T + A) = S$

Skew Symmetric: $J^T = \frac{1}{2}(A - A^T)^T = \frac{1}{2}(A^T - A) = -\frac{1}{2}(A - A^T) = -J$

$$S + J = \frac{1}{2}(A + A^T) + \frac{1}{2}(A - A^T) = \frac{1}{2}(2A) = A$$

Uniqueness: Suppose $A = S_1 + J_1 = S_2 + J_2$

where $S_1^T = S_1$, $S_2^T = S_2$, $J_1^T = -J_1$, $J_2^T = -J_2$

$S_1^T + J_1^T = S_2^T + J_2^T \rightarrow S_1 - J_1 = S_2 - J_2$

Adding, $2S_1 = 2S_2 \rightarrow S_1 = S_2$

$2J_1 = 2J_2 \rightarrow J_1 = J_2$

$\Longrightarrow$ $S = \frac{1}{2}(A + A^T)$

$J = \frac{1}{2}(A - A^T)$

5. Solving Linear Systems

a) ~~*~~ Row i of B contains the consecutive integers from $(i-1)n+1$ to in.

Let r_i be the ith row.

$$r_2 - r_1 = [n, n, n, \dots, n] = n \cdot 1$$
$$r_3 - r_2 = [n, n, n, \dots, n] = n \cdot 1$$
$$\vdots$$
$$r_n - r_{n-1} = [n, n, n, \dots, n] = n \cdot 1$$

Therefore:

$$r_2 = r_1 + n \cdot 1$$
$$r_3 = r_1 + 2n \cdot 1$$
$$r_i = r_1 + (i-1)n \cdot 1$$

Therefore for every row of B, it is a linear combination of $r_1 = [1, 2, 3, \dots, n]$ and $1 = [1, 1, 1, \dots, 1]$.

Linear independence: If $r_1 = c \cdot 1$ for some c, then $1 = 2 = 3 = \dots = n$, which is false for $n \geq 2$.

$\therefore$ r_1 and 1 are linearly independent, and:

$$\text{rank}(B) = \begin{cases} 2, & n \geq 2 \\ 1, & n = 1 \end{cases}$$

See code below.

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 5 (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 $\mathrm{rank}(B)$. 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 = [B; (i-1)*n + 1 : i*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 $[n - \mathrm{rank}(B)]$ 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
%}
alpha = 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*alpha;
disp(x);
```

```
    0.0000
   -0.0000
    0.0000
   -0.0000
    0.0000
   -0.0000
   -0.0000
```

```
-0.0000
 0.0000
-0.0000
 0.0000
 0.0000
-0.0000
 0.0000
 0.0000
-0.0000
 0.0000
-0.0000
 0.0000
-0.0000
-0.0000
 0.0000
-0.0000
 0.0000
 0.0000
 0.0000
-0.0000
-0.0000
 0.0000
-0.0000
-0.0000
-0.0000
-0.0000
-0.0000
-0.0000
 0.0000
-0.0000
-0.0000
-0.0000
 0.0000
 0.0000
-0.0000
 0.0000
 0.0000
-0.0000
 0.0000
-0.0000
-0.0000
 0.0000
 0.0000
 0.0000
-0.0000
-0.0000
-0.0000
-0.0000
 0.0000
 0.0000
-0.0000
 0.0000
 0.0000
-0.0000
 0.0000
 0.0000
 0.0000
-0.0000
 0.0000
 0.0000
 0.0000
 0.0000
 0.0000
 0.0000
```

```
    0.0000
    0.0000
    0.0000
    0.0000
   -0.0000
    0.0000
    0.0000
    0.0000
    0.0000
   -0.0000
   -0.0000
    0.0000
   -0.0000
   -0.0000
   -0.0000
    0.0000
    0.0000
   -0.0000
    0.0000
    0.0000
    0.0000
   -0.0000
   -0.0000
    0.0000
   -0.0000
    0.0000
    0.0000
   -0.0000
    0.0100
```

```
%{
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);
```

```
   3.0746e-12
```

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