11.22 *Properties of pseudo-inverses.* For an $m \times n$ matrix A and its pseudo-inverse $A^\dagger$, show that $A = AA^\dagger A$ and $A^\dagger = A^\dagger AA^\dagger$ in each of the following cases.

(a) A is tall with linearly independent columns.

(b) A is wide with linearly independent rows.

(c) A is square and invertible.

a) For a square matrix D, $D^{-1}D = DD^{-1} = I$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

Since A is tall with linearly independent rows

A cannot be right invertible, but its transpose has linearly independ rows

$\therefore$ A is left invertible with A^T $\Rightarrow$ $\therefore$ Gram matrix, AA^T is non singular

$\left.\begin{matrix} A = \langle m \times n \rangle \\ A^T = \langle n \times m \rangle \end{matrix}\right\}$ $m > n \Rightarrow \langle n \times m \rangle \langle m \times n \rangle = \langle n \times n \rangle$

∗ Pseudo-Inverse Definition for $(m \geq n)$, $A^+ = (A^TA)^{-1}A^T$

$(A^TA)^{-1} \cdot A^T = A^+ \rightarrow A^+A = I = AA^+$

$\langle n \times n \rangle \cdot \langle n \times m \rangle = \langle n \times m \rangle$

∗ For $(m \geq n)$ lin. ind. col., A^+ is left inverse of A. $[A^+A = (A^TA)(A^TA)^{-1} = I]$

$\therefore$ $A = A\underline{A^+A} = AI = A$

$A^+ = \underline{A^+A}A^+ = IA^+ = A^+$

b) A is wide with linearly independ rows. $\therefore$ A^T is right inverse of A

$\therefore$ $AA^T = I$ & is invertible, $\therefore$ $A^+ = (AA^T)^{-1}A^T$

∗ Pseudo-Inverse Definition for $(m \leq n)$, $A^+ = A^T(AA^T)^{-1}$

∗ A^+ is right inverse of A $[AA^+ = (AA^T)(AA^T)^{-1} = I]$

Show $A = \underline{AA^+}A \Rightarrow IA = A$

$A^+ = A^+\underline{AA^+} \Rightarrow A^+I = A^+$

c) A is square, so it has $\langle n \times n \rangle$ dimension

A is invertible, $\therefore$ n pivot points (ind. rows & cols.)

$\therefore$ $A^+ = A^T(AA^T)^{-1} = (A^TA)^{-1}A^T = A^{-1}$

Show: $A = AA^+A = AA^{-1}A = IA = AI = A$

$A^+ = A^{-1} = A^+AA^+ = A^{-1}AA^{-1} = IA^{-1} = A^{-1}I = A^{-1} = A^+$

5.4 A lower triangular matrix A is *bidiagonal* if $A_{ij} = 0$ for $i > j+1$: ← this is Lower Bidiagonal

$$A = \begin{bmatrix} A_{11} & 0 & 0 & \cdots & 0 & 0 & 0 \\ A_{21} & A_{22} & 0 & \cdots & 0 & 0 & 0 \\ 0 & A_{32} & A_{33} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \\ 0 & 0 & 0 & \cdots & A_{n-2,n-2} & 0 & 0 \\ 0 & 0 & 0 & \cdots & A_{n-1,n-2} & A_{n-1,n-1} & 0 \\ 0 & 0 & 0 & \cdots & 0 & A_{n,n-1} & A_{nn} \end{bmatrix}.$$

Assume A is a nonsingular bidiagonal and lower triangular matrix of size $n \times n$.

(a) What is the complexity of solving $Ax = b$?

(b) What is the complexity of computing the inverse of A?

State the algorithm you use in each subproblem, and give the dominant term (exponent and coefficient) of the flop count. If you know several methods, consider the most efficient one.

Given: A is nonsingular lower bidiagonal of size $\langle n \times n \rangle$

a) $Ax = b$: $A_{11}x_1 = b_1$, $A_{21}x_1 + A_{22}x_2 = b_2$, $A_{32}x_2 + A_{33}x_3 = b_3$

$$x_1 = \frac{b_1}{A_{11}}, \quad x_2 = \frac{b_2 - A_{21}x_1}{A_{22}}, \quad x_3 = \frac{b_3 - A_{32}x_2}{A_{33}} \xrightarrow[\text{form}]{\text{general}} x_n = \frac{b_n - A_{n,n-1}x_{n-1}}{A_{n,n}}$$

$\approx 3n$ flops (Forward Substitution)

b) $AX = I \Rightarrow X_{ii} = \frac{1}{A_{ii}}$, $X_{i+1,i} = -\frac{A_{i+1}X_{ii}}{A_{i+1,i+1}}$, $X_{i+2,i} = \frac{-A_{i+2,i+1}X_{i+1,i}}{A_{i+2,i+2}}$, $\ldots$, $X_{n,i} = \frac{A_{n,n-1}X_{n-1,i}}{A_{nn}}$

$3(n-i) - 2$ flops $\quad i = 1 \rightarrow n \quad \therefore \frac{3}{2}n^2$

(column by column)

5.6 Describe an efficient method for each of the following two problems and give its complexity.

(a) Solve

$$DX + XD = B$$

where D is $n \times n$ and diagonal. The diagonal elements of D satisfy $D_{ii} + D_{jj} \neq 0$ for all i and j. The matrices D and B are given. The variable is the $n \times n$ matrix X.

Suppose $(d_{ii} + d_{jj}) x_{ij}$, $d_{ii} + d_{jj} \neq 0$

Solving for x_{ij}, $x_{ij} = \dfrac{b_{ij}}{d_{ii} + d_{jj}}$ $i,j = 1, \dots, n$

n^2 (division), n^2 (additions) $\therefore 2n^2$ flops

$d_{ii} + d_{jj}$ are symmetric, their additions are $i > j$ n and $i \leq j$ $(n+1)$, then we divide by two, to avoid double counting. $\therefore \approx \frac{n^2}{2}$

$\therefore \frac{n^2}{2} + n^2 = \frac{3}{2} n^2$

6.1 A square matrix A is called *normal* if $AA^T = A^T A$. Show that if A is normal and nonsingular, then the matrix $Q = A^{-1}A^T$ is orthogonal.

if Q is orthogonal, then $Q^TQ = I$ should be valid.

$Q = (A^{-1}A^T)$, $Q^T = (A^{-1}A^T)^T$

$\therefore Q^TQ = (A^{-1}A^T)^T(A^{-1}A^T)$ $\qquad (A^{-1}A) = (AA^{-1})$

$= (A^{-T}(A^T)^T)(A^{-1}A^T) = A^{-T}AA^{-1}A^T$

$= AA^{-T}A^{-1}A^T = A(A^TA)^{-1}A^T$ $\qquad (A^TA) = (AA^T)$

$= \underbrace{AA^{-1}}_{I}\underbrace{A^{-T}A^T}_{I}$

$= I$

6.7 *Circulant matrices and discrete Fourier transform.* A *circulant matrix* is a square matrix of the form

$$T(a) = \begin{bmatrix} a_1 & a_n & a_{n-1} & \cdots & a_3 & a_2 \\ a_2 & a_1 & a_n & \cdots & a_4 & a_3 \\ a_3 & a_2 & a_1 & \cdots & a_5 & a_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & a_{n-2} & a_{n-3} & \cdots & a_1 & a_n \\ a_n & a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 \end{bmatrix}. \tag{6}$$

We use the notation $T(a)$ for this matrix, where $a = (a_1, a_2, \ldots, a_n)$ is the n-vector in the first column. Each of the other columns is obtained by a circular downward shift of the previous column. In matrix notation,

$$T(a) = \begin{bmatrix} a & Sa & S^2a & \cdots & S^{n-1}a \end{bmatrix}$$

where S is the $n \times n$ *circular shift* matrix

$$S = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ I_{n-1} & 0 \end{bmatrix}.$$

(a) Let W be the $n \times n$ DFT matrix:

$$W = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega^{-1} & \omega^{-2} & \cdots & \omega^{-(n-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \cdots & \omega^{-2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{-(n-1)} & \omega^{-2(n-1)} & \cdots & \omega^{-(n-1)(n-1)} \end{bmatrix} \tag{7}$$

where $\omega = e^{2\pi j/n}$. Verify that

$$WS^{k-1} = \mathbf{diag}(We_k)W, \quad k = 1, \ldots, n,$$

where S^i is the ith power of S (with $S^0 = I$), e_k is the kth unit vector (hence, We_k is column k of W), and $\mathbf{diag}(We_k)$ is the $n \times n$ diagonal matrix with We_k on its diagonal.

(b) The inverse of W is $W^{-1} = (1/n)W^H$. The expression in part (a) can therefore be written as

$$S^{k-1} = \frac{1}{n} W^H \, \mathbf{diag}(We_k)W, \quad k = 1, \ldots, n.$$

Use this to show that $T(a)$ can be factored as a product of three matrices:

$$T(a) = \frac{1}{n} W^H \, \mathbf{diag}(Wa)W. \tag{8}$$

(c) The matrix-vector product $y = Wx$ is the discrete Fourier transform of x. The matrix-vector product $y = W^{-1}x = (1/n)W^Hx$ is the inverse discrete Fourier transform. The DFT and its inverse can be computed in order $n \log n$ operations using the Fast Fourier Transform algorithm.

Use the factorization (8) to formulate a fast algorithm, with a complexity of order $n \log n$, for computing matrix-vector products $T(a)x$. The product $T(a)x$ is known as the *circular convolution* of the vectors a and x.

(d) Use the factorization in part (b) to formulate a fast algorithm, with an order $n \log n$ complexity, for solving a set of linear equations $Ax = b$ with variable x and coefficient matrix $A = T(a)$ (assuming $T(a)$ is nonsingular). Compare the speed of your algorithm with the standard method (`A \ b`), for randomly generated a and b. You can use the following code to generate a and b, and construct the circulant matrix $A = T(a)$.

```
a = randn(n, 1);
b = randn(n, 1);
A = toeplitz(a, [a(1), flipud(a(2:n))']);
```

Use the MATLAB functions `fft` and `ifft` to implement the fast algorithm. `y = fft(x)` evaluates $y = Wx$ using the Fast Fourier Transform algorithm; `y = ifft(x)` evaluates $y = W^{-1}x$.

a) $k=1$ results in S^{k-1} and $\operatorname{diag}(We_1)$ are both I.

$2 \le k \le n$, the columns are shifted circularly to the left,

$$\therefore WS^{k-1} = \begin{bmatrix} 1 & \cdots & 1 & 1 & 1 & & 1 \\ \omega^{-(k-1)} & \cdots & \omega^{-(n-1)} & 1 & \omega^{-1} & & \omega^{-(k-2)} \\ \vdots & \ddots & \vdots & 1 & \vdots & & \vdots \\ & & & 1 & & & \\ \omega^{-n(n-1)(k-1)} & \cdots & \omega^{-(n-1)} & 1 & \omega^{-(n-1)} & & \omega^{-(n-1)(k-2)} \end{bmatrix}$$

the k^{th} column of W is the vector $We_k = (1, \omega^{-(k-1)}, \omega^{-2(k-1)}, \ldots, \omega^{-(n-1)(k-1)})$

$$\operatorname{diag}(We_k)W = \begin{bmatrix} 1 & \cdots & 1 & 1 & 1 & & 1 \\ \omega^{-(k-1)} & \cdots & \omega^{-(n-1)} & \omega^{-n} & \omega^{-(n+1)} & & \omega^{-(n+k-2)} \\ \vdots & & & \vdots & \vdots & & \vdots \\ \omega^{-(n-1)(k-1)} & \cdots & \omega^{-(n-1)^2} & \omega^{-(n-1)n} & \omega^{-(n-1)(n+1)} & \cdots & \omega^{-(n-1)(n+k-2)} \end{bmatrix}$$

$\omega^{-n} = 1$, $\Rightarrow WS^{k-1} = \operatorname{diag}(We_k)W \quad k = 1, -, n$

b) column k of $T(a)$:

$S^{k-1}a = \frac{1}{n} W^H \operatorname{diag}(We_k) Wa = \frac{1}{n} W^H (We_k \circ Wa) = \frac{1}{n} W^H \operatorname{diag}(Wa) We_k$

$T(a) = (a \;\; Sa \;\; S^2a \ldots S^{n-1}a) = \frac{1}{n} W^H \operatorname{diag}(Wa)(We_1 \;\; We_2 \ldots We_n) = \frac{1}{n} W^H \operatorname{diag}(Wa) W$

c) $T(a)x = \frac{1}{n} W^H \operatorname{diag}(Wa) W = W^{-1}(\underbrace{(Wa) \circ (Wx)}_{g})$

inverse DFT of $W^{-1}g$, $\therefore$ DFT, inv DFT, then DFT again
$n \log n$ complexity

d) Wa must be non zero,

$T(a)^{-1} b = W^{-1} \underbrace{\operatorname{diag}(Wa)}_{\text{DFT}}{}^{-1} \underbrace{Wb}_{\text{DFT}}$ $\therefore$ DFT(Wb)/DFT(Wa) $\rightsquigarrow$ x = ifft(fft(b) ./ fft(a)).

6.11 Let a be an n-vector with $\|a\| = 1$. Define the $2n \times 2n$ matrix

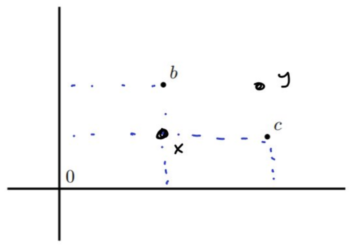
$\|a\| = a^Ta = 1$

$$A = \begin{bmatrix} aa^T & I - aa^T \\ I - aa^T & aa^T \end{bmatrix}.$$

(a) Show that A is orthogonal.

(b) The figure shows an example in two dimensions ($n = 2$). Indicate on the figure the 2-vectors x, y that solve the 4×4 equation

$$\begin{bmatrix} aa^T & I - aa^T \\ I - aa^T & aa^T \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}.$$



line through a and the origin

a) if A is orthogonal, then $A^TA = I$

$$A = \begin{bmatrix} aa^T & I - aa^T \\ I - aa^T & aa^T \end{bmatrix}$$

$$A^TA = \begin{bmatrix} aa^T & I - aa^T \\ I - aa^T & aa^T \end{bmatrix} \begin{bmatrix} aa^T & I - aa^T \\ I - aa^T & aa^T \end{bmatrix}$$

$$= \begin{bmatrix} aa^Taa^T + (I - aa^T)(I - aa^T) & aa^T(I - aa^T) + (I - aa^T)aa^T \\ (I - aa^T)aa^T + aa^T(I - aa^T) & aa^Taa^T + (I - aa^T)(I - aa^T) \end{bmatrix}$$

$$= \begin{bmatrix} aa^T + I - 2aa^T + aa^T & 2(aa^T - aa^T) \\ 2(aa^T - aa^Taa^T) & I - 2aa^T + aa^T + aa^T \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = I$$

b)

$$\begin{bmatrix} aa^T & I - aa^T \\ I - aa^T & aa^T \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix}$$

$A = A^{-1}$ b/c orthogonal

$$\to \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} aa^T & I - aa^T \\ I - aa^T & aa^T \end{bmatrix} \begin{bmatrix} b \\ c \end{bmatrix} = \begin{matrix} \overbrace{aa^Tb}^{\text{proj b on x}} + \overbrace{(I - aa^T)c}^{\text{proj c on x}} = x \\ \underbrace{(I - aa^T)b}_{\text{proj b on y}} + \underbrace{(aa^T)c}_{\text{proj c on y}} = y \end{matrix}$$

7.5 You are given a nonsingular $n \times n$ matrix A and an n-vector b. You are asked to evaluate

$$x = (I + A^{-1} + A^{-2} + A^{-3})\, b$$

where $A^{-2} = (A^2)^{-1}$ and $A^{-3} = (A^3)^{-1}$. Describe in detail how you would compute x, and give the complexity of the different steps in your algorithm. If you know several methods, give the most efficient one.

$X = (I - A^{-1} + A^{-2} + A^{-3})\, b = Ib + A^{-1}b + A^{-2}b + A^{-3}b$

$A^{-2} = (A^2)^{-1} \qquad A^{-3} = (A^3)^{-1}$

1) LU-factorization, $A = PLU \rightsquigarrow \frac{2}{3}n^3$ flops

2) $PLUy = b$

3) $v = P^{-1}b$ Permutation = 0 flops

4) $Lw = v$ Forward Sub $\simeq n^2$ flops

5) $Uy = w$ backward substitution n^2 flops

6) $v = A^{-2}b = A^{-1}y \rightsquigarrow PLUv = y \sim 2n^2$ flops

7) $w = A^{-3}b = A^{-1}v \rightsquigarrow PLUw = v \sim 2n^2$ flops

8) $x = b + y + v + w \sim 3n$ flops

$\therefore \frac{2}{3}n^3 + 6n^2 + 3n$ flops

7.9 Suppose you have to solve two sets of linear equations

$$Ax = b, \qquad (A + uv^T)y = b,$$

where A is $n \times n$, and u, v, and b are n-vectors. The variables are x and y. We assume that A and $A + uv^T$ are nonsingular. Give an efficient method, based on the expression

$$(A + uv^T)^{-1} = A^{-1} - \frac{1}{1 + v^T A^{-1} u} A^{-1} u v^T A^{-1}.$$

Clearly state the different steps in your algorithm and give their complexity.

$$Ax = b$$

$$x = A^{-1} b$$

$$y = (A + uv^T)^{-1} b = A^{-1}b - \frac{1}{1 + v^T A^{-1} u} A^{-1} u v^T A^{-1} b$$

$$= A^{-1} b - \frac{v^T A^{-1} b}{1 + v^T A^{-1} u} A^{-1} u$$

$A = PLU$ $\quad \frac{2}{3} n^3$ flops

$x = A^{-1} b \rightarrow 2n^2$ flops

$A^{-1} u \rightarrow 2n^2$ flops

$\frac{v^T A^{-1} b}{1 + v^T A^{-1} v} \rightarrow 4n$ flops

$A^{-1} b - \frac{v^T A^{-1} b}{1 + v^T A^{-1} u} A^{-1} u \rightarrow 2n$ flops

total $= \frac{2}{3} n^3 + 4n^2 + 6n \approx \frac{2}{3} n^3$