Lecture 7: Givens Orthogonalization Methods, Weighted Least Squares and Computation of Pseudo Inverse

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering University of Connecticut

Contact: krishna@engr.uconn.edu (860) 486-2890

Fall 2008

October 8, 2008

ECE 6435 Adv Numerical Methods in Sci Comp

Copyright © 2008 by K. Pattipati

Outline of Lecture 7

- **Q** Givens Transformations
- Weighted Least Squares Problem and its Solutions via Householder Transformation
- □ Computation of Pseudo (Generalized) Inverse

How do Given's Rotations Work? - 1

1

\nElectric zeroing of elements and selective revision of *R*

\n
$$
\begin{bmatrix}\nx & x & x \\
x & x & x \\
x & x & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
0 & x & x \\
x & x & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
0 & x & x \\
x & x & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
0 & x & x \\
x & x & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
0 & x & x \\
x & x & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
x & x & x \\
x & x & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
x & x & x \\
x & x & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
x & x & x \\
x & x & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
x & x & x \\
x & x & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
0 & x & x \\
0 & x & x \\
0 & 0 & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
0 & x & x \\
0 & 0 & x \\
0 & 0 & x\n\end{bmatrix}\n\begin{bmatrix}\nx & x & x \\
0 & x & x \\
0 & 0 & x \\
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & x & x\n\end{bmatrix}
$$

5 Copyright [©] 2008 by K. Pattipati

Why is Givens Transformation a Rotation?
\n
$$
\begin{bmatrix}\ny_1 \\
y_2\n\end{bmatrix} = \begin{bmatrix}\n\cos\theta & \sin\theta \\
-\sin\theta & \cos\theta\n\end{bmatrix} \begin{bmatrix}\nx_1 \\
x_2\n\end{bmatrix} \Rightarrow \text{Rotation of X-Y axis through an angle } \theta
$$
\nAlso, $\begin{bmatrix}\nx_1 \\
x_2\n\end{bmatrix} = \begin{bmatrix}\n\cos\theta & -\sin\theta \\
\sin\theta & \cos\theta\n\end{bmatrix} \begin{bmatrix}\ny_1 \\
y_2\n\end{bmatrix}$
\nSo, we have the important result that
\n $J^{-1}(1,2,\theta) = J^T(1,2,\theta) = J(1,2,-\theta)$
\n• In general, $J(i,k,\theta)$ rotates *i-k* coordinates by an angle θ in a
\ncounter-clockwise direction.
\n $J(i,k,\theta) \underline{x} = \underline{y}$
\n $\Rightarrow y_i = cx_i + sx_k, y_k = -sx_i + cx_k, y_j = x_j \forall j \neq i, k$

٠. ۰.

Householder versus Givens

• Also, note in 2 by 2 case, if **versus Givens**
 $\begin{bmatrix} v_1, v_2 \end{bmatrix}^T$ where, $v_1 = -\sin \theta$, $v_2 = \cos \theta$ *T* **ler versus Givens**
 $\underline{v} = [v_1, v_2]^T$ where, $v_1 = -\sin \theta$, $v_2 = \cos \theta$ Givens
re, $v_1 = -\sin \theta$, $v_2 = \cos \theta$
 θ $\sin 2\theta$

Also, note in 2 by 2 case, if
$$
\underline{v} = [v_1, v_2]^T
$$
 where, $v_1 = -\sin \theta$, $v_2 = \cos \theta$
\n
$$
W = \begin{bmatrix} I - 2\underline{v} \underline{v}^T \end{bmatrix} = \begin{bmatrix} 1 - 2v_1^2 & -2v_1v_2 \\ -2v_1v_2 & 1 - 2v_2^2 \end{bmatrix} = \begin{bmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{bmatrix}
$$
\nFor a 2 by 2 case

- For a 2 by 2 case $W = \begin{bmatrix} I - 2v v^T \end{bmatrix} = \begin{bmatrix} 1 - 2v_1 & -2v_1v_2 \ -2v_1v_2 & 1 - 2v_2^2 \end{bmatrix} = \begin{bmatrix} \cos 2\theta \\ \sin 2\theta \end{bmatrix}$

for a 2 by 2 case
 $Q = \begin{bmatrix} c & s \end{bmatrix}$ is a Householder (or reflection) $Q = \begin{vmatrix} 0 & 0 \\ 0 & 1 \end{vmatrix}$ or $\begin{vmatrix} 0 & 0 \\ 0 & 1 \end{vmatrix}$ is a rotation the *s* c s c s c s -c $\begin{bmatrix} s & s \\ c & s \end{bmatrix}$ is a House
 $\begin{bmatrix} c & -s \\ c & -s \end{bmatrix}$ or $\begin{bmatrix} c & -s \\ 0 & -s \end{bmatrix}$ $=\begin{bmatrix} s & -c \end{bmatrix}$ is a Householder (or re
 $=\begin{bmatrix} c & s \ -s & c \end{bmatrix}$ or $\begin{bmatrix} c & -s \ s & c \end{bmatrix}$ is a rotatio $\begin{bmatrix} I - 2 \underline{v} \underline{v}^T \end{bmatrix} = \begin{bmatrix} 1 - 2 \underline{v}^T \ -2 \underline{v}^T \end{bmatrix}$
2 by 2 case
 $\begin{bmatrix} c & s \end{bmatrix}$ is a House a 2 by 2 case
= $\begin{bmatrix} c & s \\ s & -c \end{bmatrix}$ is a House! $\begin{bmatrix} c & s \\ s & -c \end{bmatrix}$ is a Householder (or re
 $\begin{bmatrix} c & s \end{bmatrix}$ or $\begin{bmatrix} c & -s \\ s & s \end{bmatrix}$ is a rotation
- Coming back to the general case, we can force $y_k \uparrow$ to 0 by letting

Using back to the general case,

\n
$$
c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}} \quad s = \frac{x_k}{\sqrt{x_i^2 + x_k^2}}
$$

 \Rightarrow Any specified element can be zeroed out by appropriate choice of *c* and *s*

H

 \Rightarrow Since the effect is local, the procedure is well-suited for parallel processing

Implementation Issues
\n
$$
|x_k| \ge |x_i|
$$
, write $t = x_i / x_k$; $s = (1 + t^2)^{-1/2}$, $c = st$
\n $|x_i| \ge |x_k|$, write $t = x_k / x_i$; $c = (1 + t^2)^{-1/2}$, $s = ct$

I Implementation

 $(1+t^2)$ $2 \int^{-1/2}$ $2 \int^{-1/2}$ if $x_k = 0$ $c = 1$ $s = 0$ else if else if $|x_k| \ge |x_i|$
 $t = x_i / x_k$; $s = (1 + t^2)^{-1/2}$, $c = st$ else else
 $t = x_k / x_i; c = (1 + t^2)^{-1/2},$ $\begin{aligned} 0 \\ x_k \geq |x_i| \end{aligned}$ $t = x_k / x_i$; $c = (1 + t^2)^{-1/2}$, $s = ct$ $\overline{}$ \overline{a} \geq $|\geq |x_i|$
= x_i / x_k ; $s = (1 + t^2)^{-1/2}$, $c = st$ = x_k / x_i ; $c = (1 + t^2)^{-1/2}$, $s = ct$

$$
t = x_k / x_i; c = (1 + t^2)^{-1/2}, s = ct
$$

end if

g Copyright [©] 2008 by K. Pattipati

Givens Orthogonalization Procedure

Algorithm: Givens

 $(k-1,n)$ $\leftarrow J(i,k,\theta)$ ns
For *k*=2,...,*m* DO For $k=2,...,m$ DO
For $i=1,2,...,\min (k-1,n)$ DO Find c and $s \ni$ $\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_{ii} \\ a_{ki} \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}$ $\begin{bmatrix} c & s \\ -s & c \end{bmatrix}$ $\begin{bmatrix} a_{ii} \\ a_{ki} \end{bmatrix}$
 $A \leftarrow J(i,k,$ *ki*,min
c and *s* and *s* \overrightarrow{a}
 c \overrightarrow{s} \overrightarrow{a} \overrightarrow{a} \overrightarrow{a} \overrightarrow{a} $\begin{bmatrix} a \\ s \\ c \end{bmatrix}$ $\begin{bmatrix} a \\ a \end{bmatrix}$ $\begin{bmatrix} -s & c \end{bmatrix} \begin{bmatrix} a \\ a_{ki} \end{bmatrix} =$
A \leftarrow *J* (i, k, θ) *A* and $s \rightarrow$
 $\begin{bmatrix} c & s \end{bmatrix} \begin{bmatrix} a_{ii} \end{bmatrix} = \begin{bmatrix} x \end{bmatrix}$ $\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_{ii} \\ a_{ki} \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}$

End DO

- Number of operations: $2n^2$ 3 $n^2\left(m-\frac{n}{3}\right)$
- If you want to solve LS problem, insert $\underline{b} \leftarrow J(i,k,\theta)\underline{b}$ and solve $Rx = b$

Assume Full Column Rank. Suppose we have an initial solution \underline{x}_{LS}^0 Can we improve it ? YES !! Iteration *k*=0 $\mathop{\rm{ii}}\limits_{(k)}$ e improve it ? YES !!
Iteration $k=0$
Compute residual $\underline{r}^{(k)} = \underline{b} - A \underline{x}_{LS}^{(k)}$ in double precision 2 $Ax_{LS}^{(k)}$ in $(x - \underline{r}^{(k)} \|_2^2)$ b l (k) 1 \ddot{e}
(*k*) (LS | Solve LS problem min $+ \rightarrow$ Compute re
 $|$ Solve LS p
 $|$ \Rightarrow solve R
 $|$ where r is | Solve LS problem n
 \Rightarrow solve R₁ \leq ^(k) = r_c

| where r_c is given by | | |
|
|
|
| *k*) = <u>*b*</u> - *A*_{$x_{LS}^{(k)}$ i
 $\min_{z_{(k)}} ||A_{Z}(k) - \underline{r}_{(k)}||$} *k C* r_c *n* here r_C is g_C
 $T \underline{r}^{(k)} = \begin{bmatrix} r_C \\ r_C \end{bmatrix}$ $\begin{bmatrix} c \\ c \\ d \end{bmatrix}_{m-n}^{n}$ $x_{\text{LS}}^{(k+1)} = x_{\text{LS}}^{(k)}$ **resume Full Column Rank. Suppose we have an we improve it? YES !!**

Iteration $k=0$
 \rightarrow Compute residual $\underline{r}^{(k)} = \underline{b} - A \underline{x}_{LS}^{(k)}$ in double *h*
A_Z^(*k*) – *r*
A_Z^(*k*) – *r z*
idual <u>r</u>
oblem
<u>z</u>^(k) = r_e
oiven b *r* ⇒ solve R₁ \leq ^(k)
where r_c is give
 $Q^T \underline{r}^{(k)} = \begin{bmatrix} r_c \\ r_d \end{bmatrix}_{m=0}^n$ teration $k=0$
Compute residual $\underline{r}^{(k)} = \underline{b} - A \underline{x}_{LS}^{(k)}$
Solve LS problem $\min_{\underline{z}^{(k)}} ||A \underline{z}^{(k)} - \underline{r}$
 \Rightarrow solve $R_1 \underline{z}^{(k)} = r_C$
where r_1 is given by problem $\lim_{\underline{z}^{(k)}} ||A_{\underline{s}}$
 $R_1 \underline{z}^{(k)} = r_c$

is given by
 $\begin{bmatrix} r_c \\ r \end{bmatrix}^n$ e R₁ \leq ^(k) = r_c
 \therefore is given by

= $\begin{bmatrix} r_c \\ r_d \end{bmatrix}_{m-n}^n$
 $x^{(k)} + z^{(k)}$ $Q^T \underline{r}^{(k)} = \begin{bmatrix} r_C \\ r_d \end{bmatrix}_{m-n}^n$
 $\frac{x_{LS}^{(k+1)}}{r} = \frac{x_{LS}^{(k)}}{x_{LS}} + \frac{x_{LS}}{x_{LS}}$
 K if $\frac{x_{LS}}{x_{LS}}$ has converged, stop. LS *x* 2 | else $k = k + 1$ $\begin{array}{ccc}\n & \text{If } \underline{x}_{LS} \\
 & \text{else} \\
 & \text{else}\n \end{array}$ | else

| $k = k + 1$

+ → endif

Computational load: $O(mn + n^2/2)$

B by K. Pattipati r_C is given by
 $x_b = \left[\frac{r_C}{r_d}\right]_{m-n}^{n}$
 $= \frac{x_{LS}^{(k)} + \underline{z}^{(k)}}{n}$

has converged stop ⁺¹⁾ = $\underline{x}_{\text{L}}^{k}$
 \underline{k}_{L} has
 $k = k$

dif = $\underline{x}_{LS}^{(k)} + \underline{z}^{(k)}$
has converged, stop.
= $k + 1$ \rightarrow $\ddot{}$ **Iterative Improvement of LS Solution**

12 Copyright $^{\circ}$ 2008 by K. Pattipati

Optimality Conditions of WLS Problem

 \Box Better way: What are the necessary conditions of optimality ?

y: What are the necessary conditions of optimality?
\n
$$
(A^T V^{-1} A) \underline{x} = A^T V^{-1} \underline{b}
$$
\n
$$
\underline{V} = SS^T \Rightarrow V^{-1} = (S^{-1})^T S^{-1}
$$
\n
$$
\underline{x}_{LS} = (A^T V^{-1} A)^{-1} A^T V^{-1} \underline{b} \text{ when } A \text{ is full rank}
$$
\n
$$
= \left[(S^{-1} A)^T S^{-1} A \right]^{-1} (S^{-1} A)^T (S^{-1} \underline{b})
$$

To derive an efficient method, let us look at an alternate problem:

fficient method, let us look at an alternate problem:
\n
$$
\min \frac{1}{2} \underline{v}^T \underline{v}
$$
\ns.t. $\underline{b} = A \underline{x} + S \underline{v}$ (2)

Necessary Conditions of Optimality

Necessary Conditions of Optimality
\n
$$
L = \frac{1}{2} \underline{v}^T \underline{v} + \lambda^T [A \underline{x} + S \underline{v} - \underline{b}]
$$
\n
$$
\Rightarrow \partial L / \partial \underline{v} = \underline{0} \Rightarrow \nu + S^T \underline{\lambda} = \underline{0}
$$
\n
$$
\partial L / \partial \underline{x} = \underline{0} \Rightarrow A^T \underline{\lambda} = \underline{0}
$$
\n
$$
\partial L / \partial \underline{\lambda} = \underline{0} \Rightarrow \underline{b} = A \underline{x}_{LS} + S \underline{v}
$$
\n
$$
\text{or, } \underline{b} = A \underline{x}_{LS} - S S^T \underline{\lambda} \Rightarrow \underline{\lambda} = V^{-1} (A \underline{x}_{LS} - \underline{b})
$$
\n
$$
\underline{v} = S^{-1} (\underline{b} - A \underline{x}_{LS}) \text{ weighted residual or "whitened residual"}
$$

 $\frac{1}{\underline{b}} - A \underline{x}_{LS}$ and $y = S^{-1}(\underline{b} - A \underline{x}_{LS})$ weighted residual or "whitened residual" $\left(\frac{12 \text{ L}}{1 \text{ A}}\right)^{-1} A^T V^{-1}$ using $A^T \underline{\lambda} = 0$, we have $\frac{D}{T}$, we have
 $\frac{T}{V^{-1}A}$ A^{T} $A^T \underline{\lambda} =$ $\mu = S^{-1}(\underline{b} - A \underline{x}_{LS})$ weigh
g $A^T \underline{\lambda} = \underline{0}$, we have
 $\underline{x}_{LS} = (A^T V^{-1} A)^{-1} A^T V^{-1} \underline{b}$ \overline{a} we have
 ^{-1}A ⁻¹ $A^T V^{-1}b$

$$
\underline{x}_{LS} = \left(A^T V^{-1} A\right)^{-1} A^T V^{-1} \underline{b}
$$

- Problems 1 and 2 are equivalent \Rightarrow
- Valid for rank deficient case as well \Rightarrow

$$
\frac{b = A_{\underline{x}} + S_{\underline{v}}}{\begin{vmatrix} \frac{b}{2} & \frac{b}{2} & \frac{b}{2} & \frac{b}{2} \\ \frac{b}{2} & \frac{b}{2} & \frac{b}{2} & \frac{b}{2} \\ \frac{b}{2} & \frac{b}{2} & \frac{b}{2} & \frac{b}{2} \end{vmatrix}}}
$$
\n
$$
A_{\underline{x}}_{LS} = A(A^{T}V^{-1}A)^{-1}A^{T}V^{-1}\underline{b} = P\underline{b}
$$
\n
$$
P = P \text{projection matrix } (P^{2} = P)
$$

16 Copyright [©] 2008 by K. Pattipati

Solution via Householder

• Do Householder on *A* $\left[\begin{array}{c} 1 \\ 2 \end{array} \right]$; $Q = m \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ on A
 $R_1 \atop 0 \tbinom{1}{0}$; $Q = m[Q_1 \ Q_2]$; R_1 upper *T* $\frac{S}{\sqrt{\frac{R}{n}}}$ **Solution via Hous**
 Q^TA = $\binom{n}{m-n}$ $\begin{bmatrix} R_1 \\ 0 \end{bmatrix}$; $Q = m [Q_1 \ Q_2]$; R_1 \bigcup_{holde}
 $\frac{n}{m-n}$ on *A*
 R_1
 0
 n
 n
 n
 $n - n$
 $n - n$ **Solution via Householder**
Do Householder on *A*
 $\Rightarrow Q^T A = \frac{n}{m-n} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$; $Q = m[Q_1 \ Q_2]$; R_1 upper Δ lder on A
 $\binom{n}{n}$ $\begin{bmatrix} R_1 \\ 0 \end{bmatrix}$; $Q = m[Q_1]$ $n \, m-n$

• Form $Q_2^T S$ and find orthogonal transformation *P* such that **a** What does $A\underline{x} + S\underline{v} = \underline{b}$ mean ? *n n m* - *n*
 T n m - *n*
 T Q^{*T*} *SP* = *m* - *n*[0 *U*]; $P = m[P_1 \quad P_2]$; *U* upper $\Delta = Q_2^T SP_2$
 P M - *P n* $m - n$
ind orthogonal transform
0 U] ; $P = m[P_1 \t P_2]$
n $m - n$
= *h* mean ? $m - n[0]$
 n $n - n$

Form $Q_2^T S$ and find orthogonal transformation P such that
 $\Rightarrow Q_2^T SP = m - n[0 \quad U]$; $P = m[P_1 \quad P_2]$; U upper $\Delta = Q_2^T SP_2$ *n* $m-n$
orthogonal transformation *P* such to
 $[J]$; $P = m[P_1 \t P_2]$; U upper $\Delta =$
 $-n$ *n* $m-n$
mean ? $\left[\frac{p}{p}\right]$ = $\left[\frac{p}{p}\right]$ mean :
 $\left[\frac{p}{p}\right]$ = $\left[\frac{p}{p}\right]$ $\left[\frac{p}{p}\right]$ = $\left[\frac{p}{p}\$ $\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \underline{x} + \begin{bmatrix} \mathcal{L}_1 & \mathcal{L}_1 & \mathcal{L}_1 & \mathcal{L}_1 \\ 0 & U \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}$ $0 \mid \frac{1}{2} \mid 0$ so solve: $\begin{array}{c} \n\sum v = b \text{ mean ?} \\
\sum v = f R_1 \quad \int Q_1^T S P_1 \quad Q_1^T S P_2 \parallel P_1^T \n\end{array}$ $\begin{bmatrix} T_L \\ T_L \\ T_B \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \underline{x} + \begin{bmatrix} Q_1^T S P_1 & Q_1^T S P_2 \\ 0 & U \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \end{bmatrix}$ *n m* - *n*
 n m - *n*
 g^T_{*b*}</sub> $\left[\frac{b}{x}\right] = \left[R_1\right]_{x} + \left[Q_1^T SP_1 \quad Q_1^T SP_2\right] \left[P_1^T v\right]$ $Q_1^T \underline{b}$
 $Q_2^T \underline{b}$ $=$ $\begin{bmatrix} R_1 \\ 0 \end{bmatrix}$ $x + \begin{bmatrix} Q_1^T S P_1 & Q_1^T S P_2 \\ 0 & U \end{bmatrix} \begin{bmatrix} P_1^T \underline{v} \\ P_2^T \underline{v} \end{bmatrix}$ z *L* \cup *L* \cup *C D L* \cup *L* $n m-n$
+ $S_v = b$ mean ?
 $\left[Q_1^T \overline{b}\right] = \left[R_1\right]_{x+\left[Q_1^T SP_1 \quad Q_1^T SP_2\right] \left[P_1^T v\right]$ + $S_V = b$ mean ?
 $\begin{bmatrix} Q_1^T \underline{b} \\ Q_2^T \underline{b} \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \underline{x} + \begin{bmatrix} Q_1^T SP_1 & Q_1^T SP_2 \\ 0 & U \end{bmatrix} \begin{bmatrix} P_1^T \underline{v} \\ P_2^T \underline{v} \end{bmatrix}$ $\begin{array}{ccc}\n\lfloor 0 \rfloor & \lfloor 0 \rfloor & \lfloor 1 \rfloor & \lfloor 2 \rfloor \\
\vdots & & \\
= Q_2^T \underline{b} & \text{and} & \underline{v} = P_2 \underline{z}\n\end{array}$

$$
U_{\underline{z}} = Q_2^T \underline{b} \qquad \text{and} \qquad \underline{v} = P_2 \underline{z}
$$

\n
$$
R_1 \underline{x} = Q_1^T \underline{b} - (Q_1^T S P_1 P_1^T \underline{v} + Q_1^T S P_2 P_2^T \underline{v})
$$

\n
$$
= Q_1^T \underline{b} - (Q_1^T S \underline{v}) = Q_1^T \left[\underline{b} - S P_2 \underline{z} \right]
$$

17 Copyright $^\circ$ 2008 by K. Pattipati

A† : Computation of Pseudo Inverse

Solution of $A\underline{x} = \underline{b}$ for any $m \times n$ matrix of any rank *r*

• When $m > n$ and rank $(A) = n$

 $(A'A)$ ¹ $\Lambda^T h - \Lambda^{\dagger}$ nen m > n and rank(A) =
 $\underline{x}_{LS} = (A^T A)^{-1} A^T \underline{b} = A^{\dagger} \underline{b}$ \overline{a} m > n and rank(A) = n
= $(A^T A)^{-1} A^T \underline{b} = A^{\dagger} \underline{b}$

- We can always consider $m > n$. Otherwise, use A^T and note $(A^T)^{\dagger} = (A^{\dagger})^T$
- More generally, $\underline{x}_{LS} = A^{\dagger} \underline{b}$; $A^{\dagger} \sim \text{Moore-Penrose Inverse}$, an *n* **x** *m* matrix. Variously referred to as pseudo-inverse or generalized inverse
- □ *A[†]* satisfies the following four conditions, termed the **Moore-Penrose conditions**:

 $i.$ *A* $A^{\dagger}A = A$ *ii.* $A^{\dagger} A A^{\dagger} = A^{\dagger}$ *iii.* $(A^{\dagger}A) = (A^{\dagger}A)^{T}$ projection on to $R(A^{T})$ *iv.* $(AA^{\dagger}) = (AA^{\dagger})^T$ projection on to $R(A)$

Moore-Penrose Conditions

- Note that ordinary inverse satisfies Moore-Penrose conditions
- $(A^TA)⁻¹A^T$ satisfies conditions
- A full column rank \Rightarrow $A = QR$ and $A^{\dagger} = R^{-1}Q^{T}$
- We can show that all four conditions are satisfied by $A^{\dagger} = R^{-1}Q^{T}$

$$
-QRR^{-1}Q^{T}QR=QR=A
$$

$$
- R^{-1}Q^TQR R^{-1}Q^T = R^{-1}Q^T = A^{\dagger}
$$

$$
-QR R^{-1} Q^T = I_m = I_m^T
$$

$$
- R^{-1} Q^T Q R = I_n = I_n^T
$$

Gram-Schmidt procedure to compute A†

• Suppose *A* has rank *r*. Further suppose that *A* is partitioned as follows: $A = [R \ T] = [R \ RS]$ where $S = t \times (n - r)$ matrix

R has *r* independent columns, *T* has $(n-r)$ dependent columns \Rightarrow $\underline{t}_i = R\underline{s}_i$, $i = 1,2,...,n-r$

$$
=
$$
 $\underline{t}_i = R\underline{s}_i, i = 1, 2, ..., n-r$

=> Since *R* has linearly independent columns

 $R^{\dagger} = (R^T R)^{-1} R^T$ an *r* **x** *n* generalized inverse matrix

Also, $R^{\dagger}R = I_r$ an $r \times r$ identity matrix

 Fact: Generalized inverse of a general *A* is similar to the generalized inverse of [*R* | *RS*], since [*R* | *RS*] can be obtained by the permutation of the columns of *A*. That is,

Gram-Schmidt Procedure for Computing A†

A P¹ P² ... P^r = A P = [R | RS] $(AP)^{\dagger} = P^{-1}A^{\dagger} = P^{T}A^{\dagger} = P_{r}P_{r-1} ... P_{1} [R \mid RS]^{\dagger}$

 \Rightarrow *A*^{*†*} = *P*₁ *P*₂ ... *P*_{*r*} [*R* | *RS*] [†] i.e., do a row permutation on the pseudoinverse of [*R* | *RS*] in reverse order to obtain the pseudo-inverse of *A*.

How to compute [R | RS] †

These of [R | RS] in reverse order to obtain the pseudo-inverse of A.

\nto compute [R | RS]^\dagger

\nclaim:
$$
(AP)^\dagger = [R | RS]^\dagger = \frac{r}{n-r} \left[\frac{(I_r + SS^T)^{-1}}{S^T (I_r + SS^T)^{-1}} \right]^\dagger
$$

Why is this true? Look at LS Problem

When rank(A) = $r < n \Rightarrow$ underdetermined system

 \Box Less number of independent equations than unknowns.

 \Rightarrow Infinite number of solutions satisfying $A\underline{x} = \underline{b}$ $x_1 + x_2 = 1$

 $\Rightarrow J = (A\underline{x} - \underline{b})^T (A\underline{x} - \underline{b}) = 0$ for infinite number of $4x_1 + 4x_2 = 4$
J = $(A\underline{x} - \underline{b})^T (A\underline{x} - \underline{b}) = 0$ for infinite number of <u>x</u>

So, among these infinite number of $\{\underline{x}\}\$, let us pick one that has minimum norm $\left(\underline{x}^T \underline{x}\right) \stackrel{\sim}{\Rightarrow}$ minimum Euclidean length. T^{\bullet} ^{1/2} $\underline{x}^T \underline{x}$

 $4x_1 + 4x_2 = 4$

1

1 *x*

1

2 *x*

 \Box That is, solve

$$
\min_{\underline{x}} \quad J_1 = \underline{x}^T \underline{x}
$$

s.t. $AP \underline{x} = \underline{b}$

Optimality Conditions

 This is a **convex programming problem** and has a unique minimizing solution solution
 $x_1^T x_1 + x_2^T x_2$, $APx = b \implies Rx_1 + RSx_2$ is a **convex programming problem** and has a unique
imizing solution
 $x^T \underline{x} = x_1^T x_1 + x_2^T x_2$, $AP \underline{x} = \underline{b} \implies R \underline{x}_1 + R S \underline{x}_2 = \underline{b}$
where \underline{x}_1 and \underline{x}_2 are of dimensions r and $(n-r)$, respectively.
Define the Lag **Conditio**
is a convex programming problem and has a
imizing solution
 $x^T \underline{x} = x_1^T x_1 + x_2^T x_2$, $AP \underline{x} = \underline{b} \implies R \underline{x}_1 + R S \underline{x}_2 = \underline{b}$
where x and x are of dimensions x and $(n-r)$ **Example 18 and 7 and** $x_1^T x_1 + x_2^T x_2$ **,** $APx = b \Rightarrow Rx_1 + RSx_2 = x_1$ **and** x_2 **are of dimensions** *r* **and** $(n - r)$ **the Lagrangian function, Conditions**

convex programming problem and has a unique

ing solution
 $= x_1^T x_1 + x_2^T x_2$, $APx = b \Rightarrow Rx_1 + RSx_2 = b$
 $= x_1^T x_1 + x_2^T x_2$, $APx = b \Rightarrow Rx_1 + RSx_2 = b$

 $\frac{1}{1}$ $\frac{x_1}{2}$ + $\frac{x_2}{2}$
 $\frac{1}{2}$ and $\frac{x_2}{2}$,

• Define the Lagrangian function,

 $\frac{d}{dx_2}$ are of dimensions *r* and $(n-r)$, respect
agrangian function,
 $L = x_1^T x_1 + x_2^T x_2 + \lambda^T [Rx_1 + RSx_2 - b]$ $L = \underline{x}_1^T \underline{x}_1 + \underline{x}_2^T \underline{x}_2 + \underline{\lambda}^T [R \underline{x}_1 + R S \underline{x}_2 - \underline{b}]$

• From Karush-Kuhn-Tucker's necessary and sufficient conditions of $L = \underline{x}_1 \ \underline{x}_1 + \underline{x}_2 \ \underline{x}_2 + \underline{\lambda}$ [$R\underline{x}_1 + RS\underline{x}_2$
ush-Kuhn-Tucker's necessary and s
for convex problems
 $\sqrt{\partial \underline{x}_1} = 0 \Rightarrow 2\underline{x}_1 + R^T \underline{\lambda} = 0 \Rightarrow \underline{\lambda} = -2(R^T)$ *T* is necessary and
lems
 $T_{\lambda} = 0 \Rightarrow \lambda = -2(R^{T})$ *L* = $\underline{x}_1^2 \underline{x}_1 + \underline{x}_2^2 \underline{x}_2 + \underline{\lambda}^2 [\underline{R} \underline{x}_1 + R S \underline{x}_2 - \underline{b}]$
 L arush-Kuhn-Tucker's necessary and sufficity for convex problems
 $\partial L / \partial \underline{x}_1 = 0 \Rightarrow 2 \underline{x}_1 + R^T \underline{\lambda} = 0 \Rightarrow \underline{\lambda} = -2(R^T)^{-1} \underline{x}_1$

From Katus-**K**ullin-1 ucker's necessary and sufficient
optimality for convex problems

$$
\frac{\partial L}{\partial x_1} = 0 \Rightarrow 2\underline{x}_1 + R^T \underline{\lambda} = 0 \Rightarrow \underline{\lambda} = -2(R^T)^{-1} \underline{x}_1
$$

$$
\frac{\partial L}{\partial x_2} = 0 \Rightarrow 2\underline{x}_2 + S^T R^T \underline{\lambda} = 0
$$

$$
\frac{\partial L}{\partial \underline{\lambda}} = 0 \Rightarrow R\underline{x}_1 + RS\underline{x}_2 = \underline{b}
$$

$$
\underline{x}_2 = S^T \underline{x}_1 \Rightarrow \underline{x}_1 = (I_r + SS^T)^{-1} R^T \underline{b}
$$
Thus, minimum norm satisfying $A\underline{x} = \underline{b}$ is given by:
$$
\left[(I_r + SS^T)^{-1} R^T \right]_{b} = (AD)^T b
$$

$$
\Rightarrow \underline{x}_1 = (I_r + SS^T)^\top R^\top \underline{b}
$$

minimum norm satisfying $A\underline{x} = \underline{b}$ is given by

$$
\underline{x}_{LS} = \begin{bmatrix} (I_r + SS^T)^{-1} R^\dagger \\ S^T (I_r + SS^T)^{-1} R^\dagger \end{bmatrix} \underline{b} = (AP)^\top \underline{b}
$$

Pseudo Inverse Mechanization - 1

Suppose we have done the Gram-Schmidt procedure on *AP*
\ni.e.,
$$
[R \mid RS] = m[\begin{array}{cc} Q & | & 0]U \end{array}
$$

\nwhere $U = \begin{bmatrix} r & w \ 0 & I_{n-r} \end{bmatrix}$
\nCompute $U^{-1} = \begin{bmatrix} U_1 & W \ 0 & I_{n-r} \end{bmatrix} \Rightarrow X = -U_1^{-1}W$
\nSince $(AP)U^{-1} = [Q \quad 0] \Rightarrow [R \quad RS] \begin{bmatrix} U_1^{-1} & X \ 0 & I_{n-r} \end{bmatrix} = [RU_1^{-1} \quad RX + RS]$
\n $\Rightarrow R = QU_1 \Rightarrow R^* = (U_1)^{-1}Q^T$
\n $\Rightarrow X = -S$
\n \Rightarrow last $(n-r)$ columns of U^{-1} are $\begin{bmatrix} -S \ I_{n-r} \end{bmatrix}$ at the point we hit dependency
\nApply G-S to these last $(n-r)$ columns
\n $Z^T[-S^T \mid I_{n-r}] = [-Z^TS^T \mid Z^T]$

Apply G-S to th $\overline{}$

$$
Z^{T} \left[-S^{T} \mid I_{n-r} \right] = \left[-Z^{T} S^{T} \mid Z^{T} \right]
$$

$$
Z^{T} \left[-S^{T} \mid I_{n-r} \right] = \left[-Z^{T} S^{T} \mid Z^{T} \right]
$$

23 Copyright [©] 2008 by K. Pattipati

Note that the transformation automatically gets stored Also, using Sherman-Morrison-Woodbury formula $(I_{n-r} + S^T S)$ -1 $\left[\frac{1}{2} \right] = 1$ or is that the transformation automatically gets stored

or $\begin{bmatrix} -S \ I_{n-r} \end{bmatrix} Z = \begin{bmatrix} r & -SZ \ I & Z \end{bmatrix}$

Since $(-Z^T S^T Z^T) \begin{bmatrix} -SZ \ Z \end{bmatrix} = Z^T \begin{bmatrix} I_{n-r} + S^T S \end{bmatrix} Z = I_{n-r}$ (Columns are orthogonal) $\begin{bmatrix} I_{n-r} \end{bmatrix}$ $Z = \frac{1}{n-r} \begin{bmatrix} Z \\ Z \end{bmatrix}$
 $\begin{bmatrix} -S \\ Z \end{bmatrix}$
 $I_{n-r} + S^T S = (Z^T)$
 $\begin{bmatrix} I_{n-r} + S^T S = (Z^T) \end{bmatrix}$ 11
-S
^{*n*-r} $Z = \frac{r}{n-r} \left[\frac{-SZ}{Z} \right]$
 $T S^T Z^T \left[\frac{-SZ}{Z} \right] = Z^T \left[I_{n-r} + S^T \right]$ \int_{n-r} + $S^{T}S$] $Z = I_{n-r}$ $\begin{bmatrix} Z^T \end{bmatrix} \begin{bmatrix} -SZ \\ Z \end{bmatrix} = Z^T \begin{bmatrix} I_{n-r} + S^T S \end{bmatrix} Z = I_{n-r}$
 $T S = (Z^T)^{-1} Z^{-1} \Rightarrow (I_{n-r} + S^T S)^{-1} = Z Z^T$ $\begin{array}{ccc} -Z^T S^T Z^T \end{array}$ $\begin{bmatrix} Z \end{bmatrix} = Z^T \begin{bmatrix} I_{n-r} \end{bmatrix}$
 $\begin{array}{ccc} Z_{n-r} + S^T S = (Z^T)^{-1} Z^{-1} \Rightarrow (I_{n-r} \end{array}$ **Pseudo**
S $\begin{bmatrix} z \end{bmatrix}$ $z = \frac{r}{n-r} \begin{bmatrix} -SZ \end{bmatrix}$ *Z* **Pseudo**
at the transform
 $\begin{bmatrix} -S \\ I_{n-r} \end{bmatrix} Z = \begin{bmatrix} r \\ n-r \end{bmatrix} \begin{bmatrix} -S \\ Z \end{bmatrix}$ *SZ* ne transformation automatically ge
 $Z = \begin{bmatrix} r & -SZ \\ r & Z \end{bmatrix}$
 $Z^T S^T Z^T \begin{bmatrix} -SZ \\ Z \end{bmatrix} = Z^T \begin{bmatrix} I_{n-r} + S^T S \end{bmatrix} Z = I$ *Z* $\begin{bmatrix} Z = \begin{bmatrix} 1 & 0 \end{bmatrix} Z = \begin{$ \overline{a} $\begin{aligned} Z \end{aligned}$ = $Z^T \left[I_{n-r} + \right]$ ⁻¹ $Z^{-1} \Rightarrow \left(I_{n-r} + \right)$ $\begin{bmatrix} -Z^T S^T Z^T \end{bmatrix} \begin{bmatrix} Z^T Z^T \end{bmatrix} = Z^T \begin{bmatrix} I_{n-r} + S^T S \end{bmatrix} Z$
 $\begin{bmatrix} -r + S^T S = (Z^T)^{-1} Z^{-1} \Rightarrow \left(I_{n-r} + S^T S \right)^{-1} \end{bmatrix}$ **Pseudo Inverse Me**
hat the transformation automatical
 $\begin{bmatrix} -S \\ I \end{bmatrix} Z = \begin{bmatrix} r \\ r \end{bmatrix} \begin{bmatrix} -SZ \\ Z \end{bmatrix}$ **Pseudo Inverse Mechanization - 2**
hat the transformation automatically gets stored
 $\begin{bmatrix} -S \ I_{n-r} \end{bmatrix} Z = \int_{n-r}^{r} \begin{bmatrix} -SZ \ Z \end{bmatrix}$ prmation automatically
 $\begin{bmatrix} -SZ \\ Z \end{bmatrix} = Z^T \begin{bmatrix} I_{n-r} + S^T S \end{bmatrix} Z$ the transformation automatically gets stored
 $\begin{aligned}\nZ &= \frac{r}{n-r} \begin{bmatrix} -SZ \\ Z \end{bmatrix} \\
-Z^T S^T Z^T \begin{bmatrix} -SZ \\ Z \end{bmatrix} = Z^T \begin{bmatrix} I_{n-r} + S^T S \end{bmatrix} Z = I_{n-r} \text{ (Columes are)}\n\end{aligned}$ or $\begin{bmatrix} I_{n-r} \end{bmatrix} Z = \frac{1}{n-r} \begin{bmatrix} Z \end{bmatrix}$

Since $(-Z^T S^T Z^T) \begin{bmatrix} -SZ \ Z \end{bmatrix} = Z^T \begin{bmatrix} I_{n-r} + S^T S \end{bmatrix} Z = I_{n-r}$ (Columns are
 $\Rightarrow I_{n-r} + S^T S = (Z^T)^{-1} Z^{-1} \Rightarrow (I_{n-r} + S^T S)^{-1} = ZZ^T$ $(I_r + SS^T) = I_r - S(I_{n-r} + S^T S)$ + $SS^{T}\big)^{-1} = I_{r} - S(I_{n-r} + S^{T}S)^{-1}S^{T} = I_{r} - (SZ)(SZ)^{T}$
 $(I_{r} + SS^{T})^{-1} = S^{T} - S^{T}S.ZZ^{T}S^{T} = [I_{n-r} - S^{T}S(I_{n-r} + S^{T}S)^{-1}]$ $(S^{T}S + I_{n-r})$ Solution-Morrison-Woodbus-
Sherman-Morrison-Woodbus-
 $I = I = S(I_{n+1} + S^T S)^{-1} S^T = I_{n+1}$ 1 1 p† † † $\frac{1}{T}$ = ZZ^T
y formul
(SZ)(SZ) ison-Woodbury formula
 $-S^TS\big)^{-1}S^T = I_r - (SZ)(SZ)^T$
 $.ZZ^TS^T = [I_{n-r} - S^TS(I_{n-r} + S^TS)^{-1}]$ $(I_r + SS^T)^{-1} = I_r - S(I_{n-r} + S^TS)^{-1}S^T = I_r - (SZ)(SZ)^T$
 $S^T (I_r + SS^T)^{-1} = S^T - S^TSZZ^TS^T = [I_{n-r} - S^TS(I_{n-r} + S^TS)^{-1}]S^T$
 $= (S^TS + I_{n-r})^{-1}S^T = ZZ^TS^T = Z(SZ)^T$ $S^{T} (I_{r} + SS^{T})^{-1} = S^{T} - S^{T} S .ZZ^{T} S^{T}$
= $(S^{T} S + I_{n-r})^{-1} S^{T}$
Recall $(AP)^{\dagger} = \begin{bmatrix} I_{r} - (SZ)(SZ) \\ Z(SZ)^{T} R \end{bmatrix}$ $\binom{-r}{Z}$ (S)
((ZZ))
((ZZ) *T*
 T T T = $I_r - S(I_{n-r} + S^T S)^{-1} S^T = I_r - (SZ)(SZ)^T$ using Sherman-Morrison-Woodbi
 $r_r + SS^T$)⁻¹ = $I_r - S(I_{n-r} + S^TS)$ ⁻¹ $S^T = I_r$ $\begin{aligned} S S^T \big)^{-1} &= I_r - S \big(I_{n-r} + S^T S \big)^{-1} S^T = I_r - (SZ) (SZ \\ r + S S^T \big)^{-1} &= S^T - S^T S . Z Z^T S^T = [I_{n-r} - S^T S \big(I_{n-r} \big) \end{aligned}$ $T^T - S^T S . Z Z^T S^T = [I_{n-r} - S^T S (I_{n-r} - S^T S) I_{n-r}]$
 $T S + I_{n-r}$ $T S^T = Z Z^T S^T = Z (S Z)^T$ $S.\tilde{Z}$
n-*r T r* S^{T} ^{T} = $S^{T} - S^{T} S . Z Z^{T} S^{T} = [I_{n-1}]$
 $= (S^{T} S + I_{n-r})^{-1} S^{T} = Z Z^{T}$
 $= (I_{r} - (S Z) (S Z)^{T}]^{-1} R$ $I_{n-r} + S^T S = (Z^T)^{-1} Z^{-1} \Rightarrow (I_{n-r} + S^T S)^{-1} = Z Z^T$
 I, using Sherman-Morrison-Woodbury formula
 $I_r + S S^T \Big)^{-1} = I_r - S (I_{n-r} + S^T S)^{-1} S^T = I_r - (S Z) (S Z)^T$ *Sherman-Morrison-Woodbury formula*
 $(I_r + SS^T)^{-1} = I_r - S(I_{n-r} + S^TS)^{-1}S^T = I_r - (SZ)(SZ)^T$
 $S^T (I_r + SS^T)^{-1} = S^T - S^TS.ZZ^TS^T = [I_{n-r} - S^TS(I_{n-r} + S^TS)^{-1}]S$ $S S (I_{n-r} + S^T S)^{-1} S^T = I_r - (SZ) (S^T S^T - S^T S . ZZ^T S^T = [I_{n-r} - S^T S (I_{n-r} - S^T S (I_{n-r} - S^T S^T - Z^T S$ I_{n-r} $\Big)^{-1} S^T$
 $Z (SZ) (SZ)^T R$
 I_{n-1} -= $I_r - S(I_{n-r} + S^T S)^{-1} S^T = I_r - (SZ)(SZ)^T$
= $S^T - S^T S Z Z^T S^T = [I - S^T S (I + S^T S)^{-1}] S^T$ a Sherman-Morrison-Woodbury formula
 $(S^T)^{-1} = I_r - S(I_{n-r} + S^T S)^{-1} S^T = I_r - (SZ)(SZ)^T$
 $+ SS^T)^{-1} = S^T - S^T S.ZZ^T S^T = [I_{n-r} - S^T S(I_{n-r} + S^T S)^{-1}] S^T$ $\overline{}$ $\overline{}$ $\overline{}$ $\begin{aligned} & \left(S^T S S \right)^{-1} S^T = I_r - (SZ)(SZ)^T \\ & S^T S.ZZ^T S^T = [I_{n-r} - S^T S (I_{n-r} + S^T S)]^T \\ & + I_{n-r} \Big)^{-1} S^T = ZZ^T S^T = Z (SZ)^T \end{aligned}$ = $S^T - S^T S . Z Z^T S^T = [I_{n-r} - S^T S (I_{n-r} + S^T$

= $(S^T S + I_{n-r})^{-1} S^T = Z Z^T S^T = Z (S Z)^T$

= $\begin{bmatrix} I_r - (S Z) (S Z)^T \end{bmatrix}^{-1} R^{\dagger}$
 $Z (S Z)^T R^{\dagger}$ $\begin{bmatrix} S^T S + I_{n-r} \end{bmatrix}^{-1} S^T = ZZ^T S^T = Z (SZ)^T$
 $\begin{bmatrix} I_r - (SZ)(SZ)^T \end{bmatrix}^{-1} R^{\dagger}$
 $Z (SZ)^T R^{\dagger}$ 1 1 1 1 $=$ $\begin{bmatrix} (SZ)^T & R^T \ (ZZ)^T & R^T \end{bmatrix}$
 $\begin{bmatrix} I_1^{-1} & -SZ \ 0 & Z \end{bmatrix}$ (SZ) *T* $\varrho^{ \mathrm{\scriptscriptstyle T} }_{I U_1^{-1} \mathcal{Q}^{\mathrm{\scriptscriptstyle T} }}$ $\left[I_r - (SZ)(SZ)^T\right]^{-1} R^{\dagger}$
 $Z(SZ)^T R^{\dagger}$
 $\left[\begin{array}{cc} U_1^{-1} & -SZ \\ 0 & Z \end{array}\right] \left[\begin{array}{c} Q^T \\ Q^T \end{array}\right]$ $\begin{bmatrix} L_r - (SZ)(SZ)^r & R^r \ Z(SZ)^T R^{\dagger} & R \end{bmatrix}$
 $\begin{bmatrix} r & U_1^{-1} & -SZ \ 0 & Z \end{bmatrix} \begin{bmatrix} Q^T \ (SZ)^T U_1^{-1}Q \end{bmatrix}$ \overline{a} \overline{a} $\begin{bmatrix} - (SZ)(SZ)^T\ 2(SZ)^T R^\dagger\ \end{bmatrix} \begin{bmatrix} U_1^{-1} & -SZ \ 0 & Z \end{bmatrix} \begin{bmatrix} Q^T \ (SZ)^T U_1^{-1} Q^T \end{bmatrix}$ $\begin{bmatrix} I_r - (3Z)(3Z) & K \\ Z(SZ)^T R^{\dagger} & \\ r & F \begin{bmatrix} U_1^{-1} & -SZ \\ 0 & Z \end{bmatrix} \begin{bmatrix} Q^T \\ (SZ)^T U_1^{-1} Q^T \end{bmatrix}$

Summary of Procedure

Summary of Procedure
\n
$$
\begin{bmatrix}\nAP \\
I_n\n\end{bmatrix}\n\rightarrow\n\begin{bmatrix}\nR & S \\
I_r & 0 \\
0 & I_{n-r}\n\end{bmatrix}\n\begin{bmatrix}\nQ & 0 \\
U_1^{-1} & -S \\
0 & I\n\end{bmatrix}\n\begin{bmatrix}\nQ & 0 \\
U_1^{-1} & -SZ \\
0 & Z\n\end{bmatrix}
$$
\n
$$
\begin{bmatrix}\nQ & 0 \\
U_1^{-1} & -SZ \\
(SZ)^T U_1^{-1} & Z\n\end{bmatrix}\n\rightarrow\n\begin{bmatrix}\nQ & [(SZ)^T U_1^{-1} Q^T]^T \\
U_1^{-1} & -SZ \\
0 & Z\n\end{bmatrix}
$$
\nFinally,
$$
\begin{bmatrix}\nU_1^{-1} & -SZ \\
0 & Z\n\end{bmatrix}\n\begin{bmatrix}\nQ^T \\
(SZ)^T U_1^{-1} Q^T\n\end{bmatrix} = (AP)^{\dagger}
$$
\nNote that we can store A^{\dagger} in A , i.e., in the space occupied by
$$
\begin{bmatrix}\nQ & ((SZ)^T U_1^{-1} Q^T)^T\n\end{bmatrix}
$$

Note that we can store A^{\dagger} in A,i.e., in the space occupied by 1 \overline{C} ore A^{\dagger} in $A^T U_1^{-1} Q^T$ [Q | $((SZ)^T U_1^{-1} Q^T)^T$]
Finally, we can permute rows to obta
 $A^{\dagger} = P(AP)^{\dagger} = P_1 P_2 ... P_r(AP)$ \overline{a} $Q \mid ((SZ)^T U_1^{-1} Q^T)^T$
can permute rows to obta
= $P(AP)^{\dagger} = P_1 P_2 ... P_r(AP)^{\dagger}$

† P^* e can permute rows to obtain A^{\dagger}
 $= P(AP)^{\dagger} = P_1 P_2 ... P_r(AP)^{\dagger}$ Finally, we can permute rows to obtain
 $A^{\dagger} = P(AP)^{\dagger} = P_1 P_2 ... P_r(AP)^{\dagger}$ *A*

$$
\left[Q \mid ((SZ)^{T}U_{1}^{-1}Q^{T})^{T}\right]
$$

we can permute rows to obtain

$$
A^{\dagger} = P(AP)^{\dagger} = P_{1}P_{2}...P_{r}(AP)^{\dagger}
$$

$$
\Rightarrow \text{ given row } r \text{ with } q = r\text{ or } r\text{.}
$$

 \Rightarrow swap row *r* with c_r ,....,row 1 with c_1 te rows to obtain A^{\dagger}
= $P_1 P_2 ... P_r (AP)^{\dagger}$
r with c_r ,...., row 1 with c

26 Copyright [©] 2008 by K. Pattipati

Example via Householder
\nConsider
$$
\begin{bmatrix} U_1^T \\ W^T \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 2 & 1 \\ 3 & 5 \end{bmatrix}
$$

\n
$$
\underline{u}_1 = \begin{bmatrix} 4 + \sqrt{29} \\ 2 \\ 3 \\ 3 \end{bmatrix}; Z_1 = \begin{bmatrix} I - \frac{2u_1u_1^T}{u_1u_1^T} \end{bmatrix} = \begin{bmatrix} -0.7428 & -0.3714 & -0.5571 \\ -0.3714 & 0.9209 & -0.1187 \\ -0.5571 & -0.1187 & 0.8219 \end{bmatrix}
$$
\n
$$
Z_1 \begin{bmatrix} U_1^T \\ W^T \end{bmatrix} = \begin{bmatrix} -5.3852 & -3.1568 \\ 0 & 0.3273 \\ 0 & 3.9909 \end{bmatrix}
$$
\n
$$
Z_2 Z_1 \begin{bmatrix} U_1^T \\ W^T \end{bmatrix} = \begin{bmatrix} -5.3852 & -3.1568 \\ 0 & -4.0043 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \tilde{U}_1^T \\ 0 \end{bmatrix}
$$
\n
$$
\underline{w} = \tilde{U}_1^{-1} \begin{bmatrix} 8 \\ 2 \end{bmatrix} = \begin{bmatrix} -1.4856 \\ 0.6717 \end{bmatrix}
$$
\n
$$
\underline{x}_{1s} = Z_1 Z_2 \begin{bmatrix} \underline{w} \\ 0 \end{bmatrix} = \begin{bmatrix} 1.4968 \\ 0.2839 \end{bmatrix}
$$

- Best method yet to come......Lecture 12
- Reduce A to upper Δ form via Householder

$$
Q_R^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}
$$

• Reduce R to bi-diagonal form via Householder

$$
\begin{bmatrix}\n\mathbf{b} \\
\mathbf{c}\n\end{bmatrix}
$$
\nto bi-diagonal form via Householder\n
$$
Q_B^T R S_B = B_1 = \begin{bmatrix}\nd_1 & f_1 & 0 & 0 & \dots & 0 \\
0 & d_2 & f_2 & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \vdots & \vdots & \vdots & \vdots &
$$

• Zero the super-diagonal elements via a symmetric QR algorithm for Eigen values (Lecture 11) 1 elements via a

1 values (Lectu
 $(\sigma_1 \sigma_2 ... \sigma_n)$, and 0 thm for Eigen values (Lecture 11)
 $BS_{\Sigma} = \Sigma = diag(\sigma_1 \sigma_2 ... \sigma_n)$, and $B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}$
 $(Q_B^T I_{m-n}) Q_R^T A S_B S_{\Sigma} = \Sigma$; $U = Q_R (Q_B I_{m-n}) Q_{\Sigma}$; *T n* $T_{\Sigma}^{T}BS_{\Sigma} = \Sigma = d$
 *T*_{Σ} $(Q_{R}^{T}I_{m-n})Q_{R}^{T}$ *B* super-diagonal elements via a syn

gorithm for Eigen values (Lecture
 $Q_{\Sigma}^{T}BS_{\Sigma} = \Sigma = diag(\sigma_1 \sigma_2 ... \sigma_n)$, and *B* for the form and the state of $Q_{\Sigma}^{T}BS_{\Sigma} = \Sigma = diag(\sigma_{1}\sigma_{2}...\sigma_{n})$, and $B = \begin{bmatrix} B_{1} \\ 0 \end{bmatrix}$ $A^{\dagger} = \sum_{i=1}^{n} \frac{1}{2}$
 $Q_{\Sigma}^{T} (Q_{B}^{T} I_{m-n}) Q_{R}^{T} A S_{B} S_{\Sigma} = \Sigma$; $U = Q_{R} (Q_{B} I_{m-n}) Q_{\Sigma}$; $V = S_{B} S$ super-diagonal elements via a symm
rithm for Eigen values (Lecture 1
 $\sum_{\Sigma}^{T} BS_{\Sigma} = \sum = diag(\sigma_1 \sigma_2 ... \sigma_n)$, and $B =$ diagonal elements via a symmetric

of the form of the contract of the same state a_n
 $\begin{bmatrix}\n1 & 0 & \cdots & \cdots & \cdots & a_n\n\end{bmatrix}$
 $= \sum = diag(\sigma_1 \sigma_2 ... \sigma_n)$, and $B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}$ $A^{\dagger} = \sum_{i=1}^n \frac{y_i u_i^2}{\sigma_i}$ all the subset of $B = \begin{bmatrix} B_1 \ 0 \end{bmatrix}$ $A^{\dagger} = \sum_{i=1}^n \frac{v_i u_i^T}{\sigma_i}$
= Σ ; $U = Q_R(Q_B I_{m-n})Q_{\Sigma}$; $V = S_B S_{\Sigma}$

$$
B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \qquad A^{\dagger} = \sum_{i=1}^{n} \frac{\underline{v}_i \underline{u}_i^T}{\sigma_i}
$$

 $A = U \Sigma V^T \Rightarrow A^{\dagger} = V \Sigma^{\dagger} U^T$

$$
Q_{\Sigma}^{T}(Q_{B}^{T}I_{m-n})Q_{R}^{T}AS_{B}S_{\Sigma} = \Sigma ; U = Q_{R}(Q_{B}I_{m-n})Q_{\Sigma} ; V = S_{B}S_{\Sigma}
$$

Iterative Improvement of Inverse

Given $X_0 = A_0 \sim A^{-1}$, find X_1 better than X_0 \Box The method is based on Newton's method for solving $f(x) = 0$ $=$ $> x_{n+1} = x_n - f(x_n)/f'(x_n)$ \Box Applying the formula to $f(x) = a - \frac{1}{x}$ (scalar) to get \Box Typically requires $2n^3$ MADDS / iteration (expensive) $x_{n+1} = x_n - \left[a - 1/x_n\right] / \left[1/x_n^2\right] = x_n + x_n(1 - ax_n) = x_n + (1 - x_n a)$

So, $x_{n+1} = x_n + e_n x_n$; e_n = error at iteration *n*.

Extending to matrices
 $X_{n+1} = X_n + (I - X_n A)X_n = E_n X_n$; X_0 = intial estimate
 $E_{n+1} = I - X_{n+1}A = I - X_n A - X_n A + (X_n$ $(I-X_nA)$ 2 1 So, $x_{n+1} = x_n + e_n x_n$; e_n = error at iteration *n*. ling to matrices
 $_1 = X_n + (I - X_nA)X_n = E_nX_n$; X_0 $X_{n+1} = I - X_{n+1}A = I - X_nA - X_nA + (X_nA)^2 = (I - X_nA)^2$
 $X_{n+1} = I - X_{n+1}A;$ $E_n = I - X_nA$ $X_{-1} = X_{n} + (I)$
 $X_{n+1} = I - X_{n+1}$ nethod is based on Newton's method for solving $f(x) = 0$
 $\sum_{n+1} = x_n - f(x_n)/f'(x_n)$

ying the formula to $f(x) = a - 1/x$ (scalar) to get
 $\sum_{n+1} = x_n - [a-1/x_n]/[1/x_n^2] = x_n + x_n(1 - ax_n) = x_n + (1 - x_n a)x_n$ Extending to matrices $x_{n+1} = x_n - [a-1/X_n] / [1/X_n] = x_n + x_n (1 - ax_n) = x_n + (1 -$

So, $x_{n+1} = x_n + e_n x_n$; e_n = error at iteration *n*.

Extending to matrices
 $X_{n+1} = X_n + (I - X_n A) X_n = E_n X_n$; X_0 = intial estimate So, $x_{n+1} = x_n + e_n x_n$; e_n = error at iteration *n*.

Extending to matrices
 $X_{n+1} = X_n + (I - X_n A)X_n = E_n X_n$; X_0 = intial es
 $E_{n+1} = I - X_{n+1}A = I - X_n A - X_n A + (X_n A)^2 = (I - I - I_n A)E_n$ $= x_n - [a - 1 / x_n] / [$
 $x_{n+1} = x_n + e_n x_n; e_n$ ding to matrices
 $n_{n+1} = X_n + (I - X_n A) X_n = E_n X_n$ $X_{n+1} = X_n + (I - X_nA)X_n = E_nX_n$; $X_0 = Y_n$
 $X_{n+1} = I - X_{n+1}A = I - X_nA - X_nA + (X_n)$ method is based on Newton's method for solving $f(x) = 0$
 $x_{n+1} = x_n - f(x_n)/f'(x_n)$

lying the formula to $f(x) = a - 1/x$ (scalar) to get
 $x_{n+1} = x_n - [a - 1/x_n]/[1/x_n^2] = x_n + x_n(1 - ax_n) = x_n + (1 - x_n a)x$ $x_1 = x_n - f(x_n)/f'(x_n)$
 x a g the formula to $f(x) = a - 1/x$ (sca
 $x_n - [a - 1/x_n]/[1/x_n^2] = x_n + x_n(1-x_{n+1} = x_n + e_n x_n; e_n$ = error at iteration *n* $X_{n+1} = X_n - [a-1/X_n] / [1/X_n] = X_n + X_n$ (1
 $X_{n+1} = X_n + e_n X_n$; e_n = error at iteration

tending to matrices
 $X_{n+1} = X_n + (I - X_n A) X_n = E_n X_n$; X $x_{n+1} = x_n + e_n x_n$; e_n =error at iteration *n*.

tending to matrices
 $X_{n+1} = X_n + (I - X_n A) X_n = E_n X_n$; X_0 = intial est
 $E_{n+1} = I - X_{n+1} A = I - X_n A - X_n A + (X_n A)^2 = (I - I - X_n A)$. $\ddot{}$ $\ddot{}$ $X_{n+1} = X_n + (I - X_n A)$
+1 = $I - X_{n+1}A = I$ hod is based on Newton's method for solving $f(x) = 0$
 $y = x_n - f(x_n)/f'(x_n)$

g the formula to $f(x) = a - 1/x$ (scalar) to get
 $= x_n - [a - 1/x_n]/[1/x_n^2] = x_n + x_n(1 - ax_n) = x_n + (1 - x_n a)x_n$
 $y = x_n + a_n x$: e -error at iteration n $x_n - f(x_n)/f'(x_n)$

ne formula to $f(x) =$
 $(-[a-1/x_n]/[1/x_n^2])$
 $= x_n + e_n x_n$; e_n = errors $[-[a-1/X_n]/[1/X_n^+] = x_n$
= $x_n + e_n x_n$; e_n = error at ite
ng to matrices
= $X_n + (I - X_n A) X_n = E_n X$ = $x_n + e_n x_n$; e_n =error at iteration *n*.

ng to matrices

= $X_n + (I - X_n A)X_n = E_n X_n$; X_0 = intial estimate

= $I - X_{n+1}A = I - X_n A - X_n A + (X_n A)^2 = (I - X_n A)^2$ 2 1 \Rightarrow Rapid convergence provided $||I - X_0A|| < 1$. $E_{n+1} = I - X_{n+1}A;$ $E_n = I - X_nA$ Quadratic convergence $E_{n+1} = I - X_{n+1}A = I - X_nA -$
 $E_{n+1} = I - X_{n+1}A;$ $E_n = I - X_n$ $I - X$
 $I_{n+1} = E_n^2$ ate X_nA tending to matrices
 $X_{n+1} = X_n + (I - X_n A)X_n = E_n X_n$; $X_0 =$ in
 $E_{n+1} = I - X_{n+1}A = I - X_n A - X_n A + (X_n A)E_{n+1} = I - X_{n+1}A;$
 $E_n = I - X_n A$ $\begin{aligned} E_{+1} - \Delta_n^{-1} \\ E_{+1} = I - \lambda \\ E_{n+1} = E \end{aligned}$ $E_{n+1} = I - X_{n+1}A = I - X_nA - X_nA + (X_nA)^2 = (I - X_nA)^2$
 $E_{n+1} = I - X_{n+1}A; \quad E_n = I - X_nA$
 $\Rightarrow E_{n+1} = E_n^2$
 \Rightarrow Rapid convergence provided $||I - X_0A|| < 1$. $I_{n+1} = I - X_{n+1}A = I -$
+1 = $I - X_{n+1}A$; E_n $^{+}$ \overline{a} $X_{n+1} = X_n + (I - X_n A)X$
 $E_{n+1} = I - X_{n+1}A = I - X_n$
 $E_{n+1} = I - X_{n+1}A;$ $E_n = X_n$
 $E_{n+1} = E_n^2$

- **Q** Givens Transformations
	- − local effect
	- − parallelization
- □ Weighted Least Squares Problem and its Solutions via Householder Transformation
- □ Computation of Pseudo (Generalized) Inverse
	- − Gram-Schmidt
	- − Householder
	- − SVD