Table of Contents

Housekeeping

```
clear all; close all; clc

%{
    CSCI HW8 main script
    Author: Connor O'Reilly
    Date: 11/5/2021
    email: coor1752@colorado.edu
%}
```

Downloading and Storing data

```
mat = readmatrix('C:\Users\corei\OneDrive\Documents\Classes
\CSCI_3656\mat1-2.txt');

%create cell array containing A matrices
%get size of A
size_mat = size(mat);

%initialize size
A_cell = cell(1 , size_mat(2));

%fill cell

for i = 1:size_mat(2)

    A_cell(i) = mat2cell( mat(: , 1 : i), size_mat(1), i );

end
```

Part 1

```
% obatain size, rank and condition number for all A_k
```

```
%could make more effiecient but low on time lol

%initialize
size_ak = zeros(26 , 2);
rank_ak = zeros(26 , 1);
cond_ak = rank_ak;

for i = 40:65
    curr = cell2mat( A_cell(i) );
    size_ak(i-39, :) = size( curr );
    rank_ak(i-39) = rank(curr);
    cond_ak(i-39) = cond(curr);
end
```

Part 2 Initialization

```
%Generate 100 random vectors for each b_i e R^m

%initilize
b_cell = cell(26, 100);

%heavy i know
j = 1;
cnt = 1;
max_a = 100 * max(mat , [] , 'all');
min_a = 100 * min(mat , [] , 'all');

for i = 1:2600
    %create random vector and store into array, lets get creative
    bmat = rand( size_mat(1) , 1);
    b_cell(j , cnt) = mat2cell( bmat , size_mat(1) , 1);
    if(cnt == 100)
        j = j + 1;
        cnt = 0;
    end
    cnt = cnt + 1;
end
```

Part 2 a

```
    %Using built-in equation solver, linsolve compute the least-
squares
    %minimizer given A_k and b_i

    %gonna need another cell array?

    %initialize
    x_true = cell(26,100);
    j = 1;
    cnt = 1;
    for i = 1:2600
        x_true(j,cnt) = mat2cell(linsolve( A_cell{j+39} , b_cell{ j,
 cnt } ),  j + 39 , 1 );
```

```
        if(cnt == 100)
            j = j + 1;
            cnt = 0;
        end
        cnt = cnt + 1;
    end
```

Part 2 b

```
    %using normal equation solver located in function section least
 squares
    %minimizer is computed using normal equations

    % i know too many for loops in this code

    %initialize
    x_ne = cell(26,100);
    err_ki_NE = zeros(26,100);
    j = 1;
    cnt = 1;
    for i = 1:2600
        %least squares minimizer
        x_ne( j , cnt) = mat2cell(method_NormEq_Cholesky( A_cell{j
+39} , b_cell{ j, cnt } ),  j + 39 , 1 ) ;

        %relative error
        err_ki_NE(j,cnt) = norm(x_ne{j,cnt} - x_true{j,cnt}) /
 norm(x_true{j,cnt});
        if(cnt == 100)
            j = j + 1;
            cnt = 0;
        end
        cnt = cnt + 1;
    end
```

Part 2 c

```
%using normal equation solver located in function section least
 squares
%minimizer is computed using normal equations

% i know too many for loops in this code

%initialize
x_qr_mine = cell(26,100);
x_qr_matlab = cell(26,100);
err_ki_QR_bad = zeros(26,100);
err_ki_QR_good = err_ki_QR_bad;
j = 1;
cnt = 1;
for i = 1:2600
    x_qr_mine( j , cnt) = mat2cell(method_ThinQR( A_cell{j+39} ,
 b_cell{ j, cnt }, 1 ),  j + 39, 1  ) ;
```

```
   x_qr_matlab( j , cnt) = mat2cell(method_ThinQR( A_cell{j+39} ,
 b_cell{ j, cnt }, 0 ),  j + 39, 1  ) ;
   %relative error
   err_ki_QR_bad(j,cnt) = norm(x_qr_mine{j,cnt}-x_true{j,cnt}) /
 norm(x_true{j,cnt});
   err_ki_QR_good(j,cnt) = norm(x_qr_matlab{j,cnt}-x_true{j,cnt}) /
 norm(x_true{j,cnt});
   if(cnt == 100)
       j = j + 1;
       cnt = 0;
   end
   cnt = cnt + 1;
end
```

Part 3

```
%for each of QR and Normal equations compute the average error over
 all the
%bi

%initialize
errk_avg_NE = zeros(24,1);
errk_avg_QR_mine = errk_avg_NE;
errk_avg_QR_matlab = errk_avg_NE;
%excluding k = 64 and 65 due to the matrices not being sym pos def

for i = 1:26
    errk_avg_NE(i) = mean(err_ki_NE(i,:));
    errk_avg_QR_mine(i) = mean(err_ki_QR_bad(i,:));
    errk_avg_QR_matlab(i) = mean(err_ki_QR_good(i,:));
end
```

Part 4

```
%plotting done in plotting section
```

Display

```
%Part 1
fprintf('\n-------------------------------------------------------------------
\n')
fprintf('Problem 1: ')
fprintf('\n-------------------------------------------------------------------
\n\n')

fprintf('      A_k        |      Size (M x N)      |      Rank      |
 Condition Number [ K ]      \n')
fprintf('-----------------------------------------------------------------------
\n')
for i = 1 : 26
```

```
    fprintf('  A_%i          |        %i x %i              |       %i
       |       %0.4f       |\n', (i+39), size_ak(i,1), size_ak(i,2),
 rank_ak(i), cond_ak(i));

 fprintf('------------------------------------------------------------------------
\n')
end

%explanation
fprintf('\n-------------------------------------------------------------------
\n')
fprintf('Discussion: ')
fprintf('\n-------------------------------------------------------------------
\n\n')

%1: the relationship between the error using QR versus the normal
 equations
fprintf('1:\n Average error increases for both QR and the normal
 equations as k increases. Least squares error using normal equations
 increases\n more drastically than the error using thin qr even with
 the for k = 64 and k = 65 being ignored\n\n');
%2: What is the relationship between the errors and the condition
 number of Ak?
fprintf('2:\nas stated before, as the condition number of A_k
 increases the least squares error also increases \n\n')
%3:  Suppose your matrix A is ill-conditioned. Which method is more
 favorable?
fprintf('3:\n looking at the matlab impolementation of thin QR vs
 NE thin QR is more favorable for ill-conditioned matrices.\n as the
 condition number increases there is noticiable round off error using
 cholesky factorization. \nFor this homeowrk, cholesky factorization
 could not be used for matrices A_64 and A_65 due to them not being
 sym pos definite. ');


-----------------------------------------------------------------
Problem 1:
-----------------------------------------------------------------

     A_k        |      Size (M x N)      |      Rank      |      Condition
 Number [ K ]
------------------------------------------------------------------------------
  A_40         |       101 x 40             |       40           |
  74.8767        |
-----------------------------------------------------------------------
  A_41         |       101 x 41             |       41           |
  103.8004       |
-----------------------------------------------------------------------
  A_42         |       101 x 42             |       42           |
  152.2856       |
-----------------------------------------------------------------------
  A_43         |       101 x 43             |       43           |
  217.5604       |
-----------------------------------------------------------------------
```

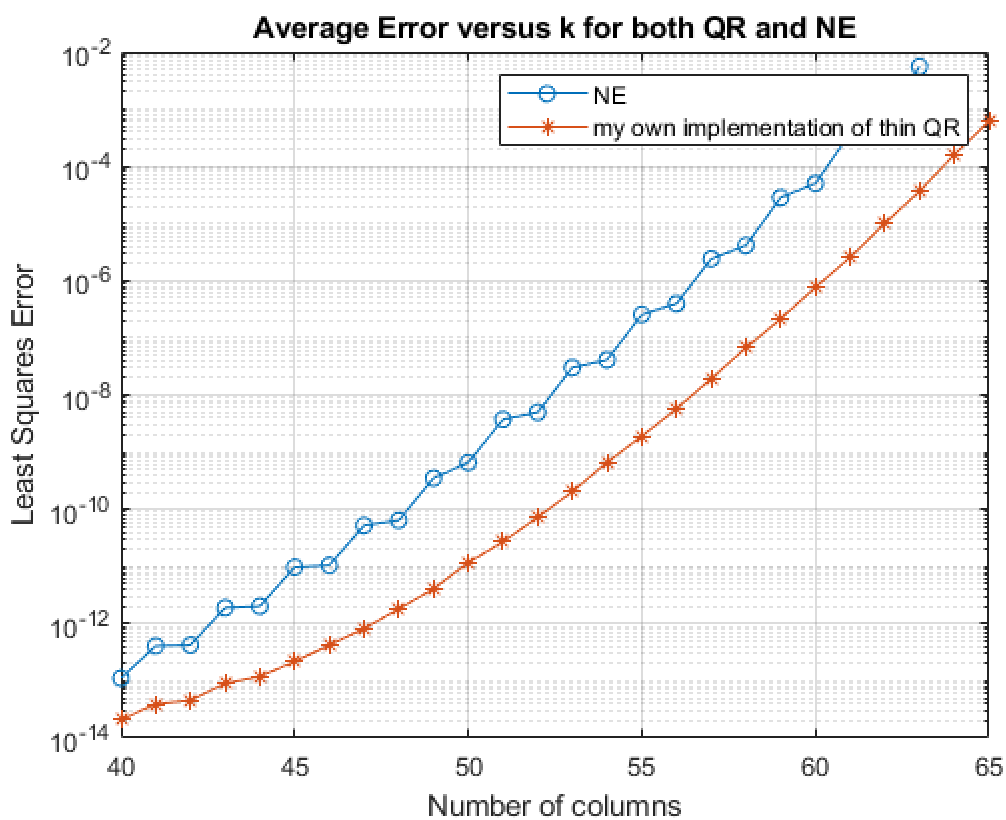
A_44	101 x 44	44	328.8920
A_45	101 x 45	45	483.7805
A_46	101 x 46	46	753.0465
A_47	101 x 47	47	1140.0742
A_48	101 x 48	48	1826.7931
A_49	101 x 49	49	2846.4223
A_50	101 x 50	50	4695.0874
A_51	101 x 51	51	7530.5483
A_52	101 x 52	52	12789.3765
A_53	101 x 53	53	21122.7169
A_54	101 x 54	54	36949.4832
A_55	101 x 55	55	62868.3337
A_56	101 x 56	56	113329.3575
A_57	101 x 57	57	198770.6821
A_58	101 x 58	58	369475.9187
A_59	101 x 59	59	668493.2863
A_60	101 x 60	60	1282274.0197
A_61	101 x 61	61	2395303.2393

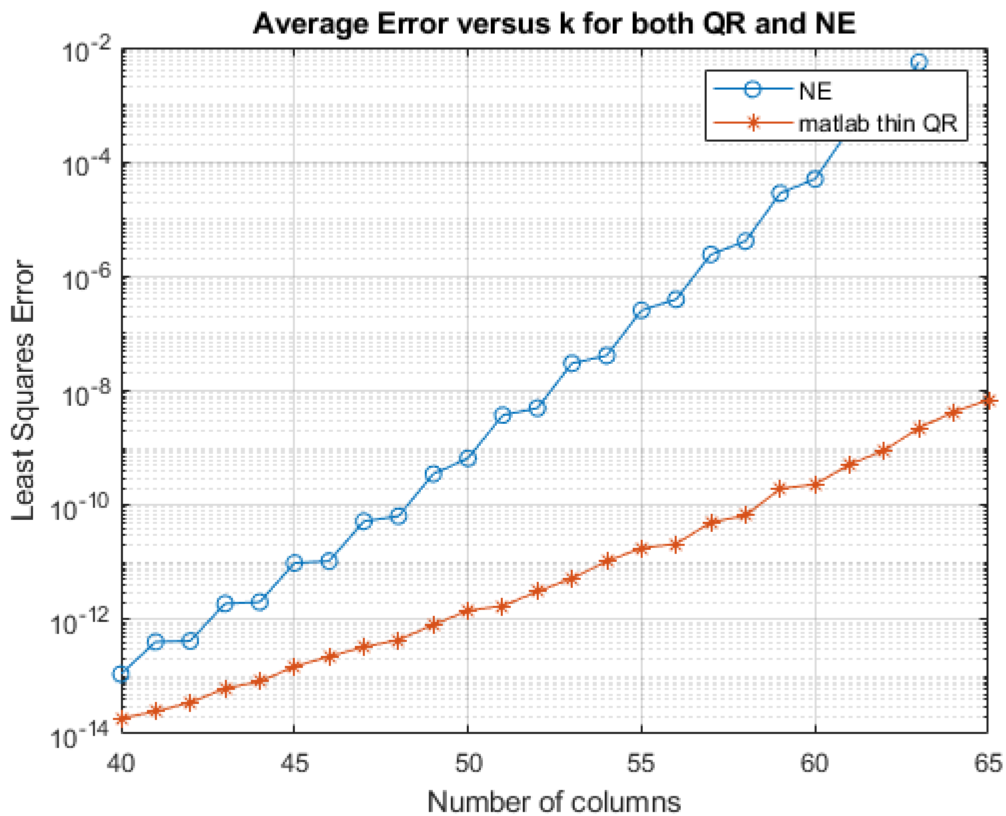
```
 A_62          |       101 x 62             |       62          |
 4745459.9263       |
-----------------------------------------------------------------------
 A_63          |       101 x 63             |       63          |
 9161533.4668       |
-----------------------------------------------------------------------
 A_64          |       101 x 64             |       64          |
 18765737.9716       |
-----------------------------------------------------------------------
 A_65          |       101 x 65             |       65          |
 37486287.3234       |
-----------------------------------------------------------------------

-----------------------------------------------------------------
Discussion:
-----------------------------------------------------------------

1:
 Average error increases for both QR and the normal equations as k
 increases. Least squares error using normal equations increases
 more drastically than the error using thin qr even with the for k =
 64 and k = 65 being ignored

2:
as stated before, as the condition number of A_k increases the least
 squares error also increases

3:
 looking at the matlab impolementation of thin QR vs NE thin QR is
 more favorable for ill-conditioned matrices.
 as the condition number increases there is noticiable round off error
 using cholesky factorization.
For this homeowrk, cholesky factorization could not be used for
 matrices A_64 and A_65 due to them not being sym pos definite.
```

Plotting

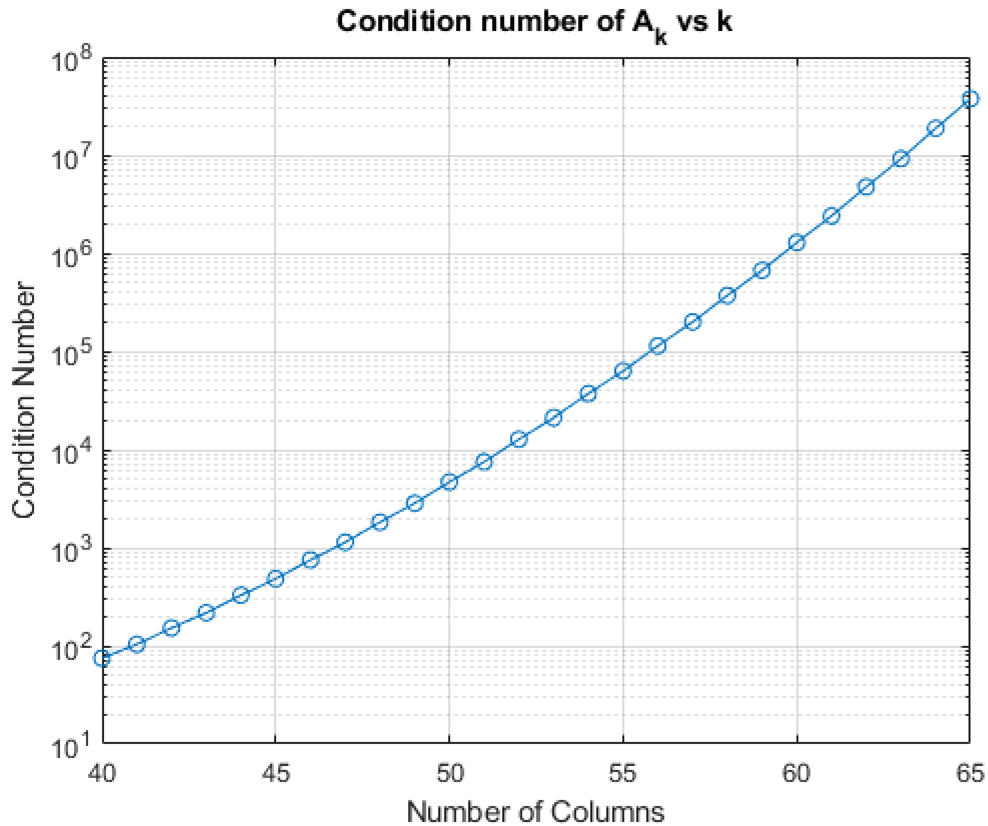
```
%average error versus k for both QR and Normal Equations using
 semilogy

%my implementation of thin QR
figure(1)
semilogy(40:65, errk_avg_NE, '-o', 40:65, errk_avg_QR_mine,'-*')
hold on;
title('Average Error versus k for both QR and NE')
ylabel('Least Squares Error')
xlabel('Number of columns')
legend('NE' , 'my own implementation of thin QR');
grid on;
hold off;

%matlab implementation of thin QR
figure(2)
```

```
semilogy(40:65, errk_avg_NE, '-o', 40:65, errk_avg_QR_matlab,'-*')
hold on;
title('Average Error versus k for both QR and NE')
ylabel('Least Squares Error')
xlabel('Number of columns')
legend('NE' , 'matlab thin QR');
grid on;
hold off;

%condition number  of A_k vs k
figure(3)
semilogy(40:65, cond_ak,'-o');
hold on;
grid on;
title('Condition number of A_k vs k')
ylabel('Condition Number')
xlabel('Number of Columns')
hold off;
```





Average Error versus k for both QR and NE
NE
matlab thin QR
Least Squares Error
Number of columns
10-2
10-4
10-6
10-8
10-10
10-12
10-14
40
45
50
55
60
65



Condition number of Ak vs k
Condition Number
Number of Columns
108
107
106
105
104
103
102
101
40
45
50
55
60
65

Functions

```
function [x] = method_NormEq_Cholesky(A,b)
%{
    Prupose: Matlab implementation of method to solve least squares
 problem using normal
    equations with cholesky
    Inputs:
        A: LHS matrix for Ax = b
        b: RHS matrix for Ax = b
    output:
        x: variable matrix x in Ax = b  (x) which minimizes
        ||b-Ax||_2^2
%}

%form A^T*A e R^n*n and A^T*b e R^n
    ata = A'*A;
    atb = A'*b;

%Solve normal eqns, (A^T*A)x = A^T*b for x
%first lets double check
%get size to make sure A = n x n
    size_A = size(ata);
%check rank
    r_ata = rank(ata);

    if( (size_A(1) == size_A(2) ) && ( r_ata == size_A(1) )  )
        %matrix is sym, pos def, compute and solve cholesky
 factorization (
        %determine variable vector using cholesky factorization
        R = chol(ata);
        x = R\(R'\atb);

    else
        %cant use error func for this homework so itll break, if using
        %outside of hw uncomment
        %error('Matrix is not Symmetric Positive Definite')

        %if this error occurs create nan array maybe later could be
 used as a flag in
        %error calc
        siz_atb = size(atb);
        warning('Error not accounted for due to A_%i'' x A is not pos
 def', siz_atb(1))
        x = NaN(siz_atb);
    end

end


function [ x ] = method_ThinQR(A, b, bool)
%{
    Purpose: Matlab implementation to solve the linear least-squares
```

```
    problem using a method based on Thin QR factorization

    Input:
        A: LHS matrix, Ax = b
        b: RHS matrix, Ax = b
        bool: boolean value, 1 if you want to use my janky code which
 seems
        wrong, or built in qr function that works
    Output:
        x: variable matrix x in Ax = b  (x) which minimizes
        ||b-Ax||_2^2
%}

%alright so because i am trying to do both the homeowkrs this weekend
 im
%assuming the grant-schmidt orthogonalization and least squares is
 similar
%to that of the thin QR decomp in lecture. Sauer's numerical analysis
 and
%https://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-
spring-2010/related-resources/MIT18_06S10_gramschmidtmat.pdf
%were used as a reference

%compute the thin QR decomposition


    if bool
        %obtain size of A
            size_A = size(A);

        %initialize r matrix. upper right traingular n x n
            r = zeros(size_A(2),size_A(2));
        %initialize q tall matrixx orthonormal columns, m x n
            q = zeros( size_A );

        %Grant-Schmidt orthogonalization
            for j = 1 : size_A(2)

                y = A(:,j);

                for i = 1 : j-1

                    r(i,j) = q(: , i)' * A(: , j);
                    y = y - r(i,j) * q(: , i);

                end
                r(j , j )= norm(y);
                q(:,j) = y/r(j , j);

            end
    else

        % if this doesnt work just use qr matlab
        [q,r] = qr(A, 0);
```

```
    end
    % After thin QR is computed, finish computation
    b_til = q'*b;

    % Solve
    x = r\b_til;

end

Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
```

```
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_64' x A is not pos def
```

```
Warning: Error not accounted for due to A_64' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
```

```
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
Warning: Error not accounted for due to A_65' x A is not pos def
```

Published with MATLAB® R2021a