Pre-Lab

1.

```matlab
% Optics 211
% Lab 3
% Prepared by Ezra A-K
% 2/29/24
%
% This script is designed to fill the workspace with the matrices necessary
% to complete the in-class portion of lab 3.  When the script is run via
% the command window, it will not yield any results or answers; however,
% the workspace will be filled with matrices.  These matrices can be called
% up from the command window for further manipulation.
%
% Appendix 1 of lab 3 serves as a companion to this script and shows all
% the matrices generated here in traditional mathematical formatting.
%
% This script can be found in Appendix 2 of lab 3.


% Part 1 of 2 of lab 3

% 5x5 matrix for data manipulation
A51= reshape((1:25), [5,5])';

% 4x4 matrix for computing the inner product and finding determinant and
% inverse
A41= [[1 2 3 4];[2 4 7 11];[3 7 14 25];[4 11 25 50]] ;

% 1x4 and 4x1 matrices for evaluating the inner product and using transpose
% operator
B41= [5;10;15;20];
B42= [5 10 15 20];

% 2x2 Matrix (invertible)
A21= [2 3;5 8];

% 2x2 Matrix (non-invertible)
A22= [6 3;8 4];


% Part 2 of 2 of lab 3

% 2 equation, 2 unknown, linear system of equations with 1 solution
A23= [2 6;-5 4];
B23= [10;-3];

% 3 equation, 3 unknown, linear system with 1 solution
A31= [1 2 3;1 3 2;3 2 1];
B31= [39;34;26];

% 3 equation, 3 unknown, linear system with an infinite number of solutions
A32= [2 4 6;4 5 6;7 8 9];
B32= [0;3;6];
```

```
% 3 equation, 3 unknown, linear System with no solution
A33= reshape((1:9), [3,3])';
B33= [0;3;0];

% End of script
```

2.

```
% Optics 211
% Lab 3
% Prepared by Ezra A-K
% 2/29/24
%
% This script is designed to fill the workspace with the matrices necessary
% to complete the in-class portion of lab 3.  When the script is run via
% the command window, it will not yield any results or answers; however,
% the workspace will be filled with matrices.  These matrices can be called
% up from the command window for further manipulation.
%
% Appendix 1 of lab 3 serves as a companion to this script and shows all
% the matrices generated here in traditional mathematical formatting.
%
% This script can be found in Appendix 2 of lab 3.


% Part 1 of 2 of lab 3

% 5x5 matrix for data manipulation
A51= reshape((1:25), [5,5])';

% 4x4 matrix for computing the inner product and finding determinant and
% inverse
A41= [[1 2 3 4];[2 4 7 11];[3 7 14 25];[4 11 25 50]] ;

% 1x4 and 4x1 matrices for evaluating the inner product and using transpose
% operator
B41= [5;10;15;20];
B42= [5 10 15 20];

% 2x2 Matrix (invertible)
A21= [2 3;5 8];

% 2x2 Matrix (non-invertible)
A22= [6 3;8 4];


% Part 2 of 2 of lab 3

% 2 equation, 2 unknown, linear system of equations with 1 solution
A23= [2 6;-5 4];
B23= [10;-3];

% 3 equation, 3 unknown, linear system with 1 solution
A31= [1 2 3;1 3 2;3 2 1];
```

```matlab
B31= [39;34;26];

% 3 equation, 3 unknown, linear system with an infinite number of solutions
A32= [2 4 6;4 5 6;7 8 9];
B32= [0;3;6];

% 3 equation, 3 unknown, linear System with no solution
A33= reshape((1:9), [3,3])';
B33= [0;3;0];

% End of script
```

3.

```
A51


A51 =


     1     2     3     4     5
     6     7     8     9    10
    11    12    13    14    15
    16    17    18    19    20
    21    22    23    24    25


>> A41


A41 =


     1     2     3     4
     2     4     7    11
     3     7    14    25
     4    11    25    50


>> B41
```

```
B41 =


     5
    10
    15
    20


>> B42


B42 =


     5    10    15    20


>> A21


A21 =


     2     3
     5     8


>> A22


A22 =


     6     3
     8     4


>> A23


A23 =
```

```
     2        6
    -5        4


>> B23


B23 =


    10
    -3


>> A31


A31 =


     1        2        3
     1        3        2
     3        2        1


>> B31


B31 =


    39
    34
    26


>> A32


A32 =
```

```
    2       4       6
    4       5       6
    7       8       9


>> B32


B32 =


    0
    3
    6


>> A33


A33 =


    1       4       7
    2       5       8
    3       6       9


>> A33


A33 =


    1       4       7
    2       5       8
    3       6       9
A33
```

```
A33 =

        1       2       3

        4       5       6

        7       8       9


>> B33


B33 =

        0

        3

        0


>>
```

Bonus

`checkcode('matrices.m')` maybe? That checks there aren't any programing errors though to check if matrix are the same you could use the isequal() command but I don't have anything to compare them to.

Part 1 A

```
>> matrices
>> A51


A51 =

        1       2       3       4       5

        6       7       8       9      10

       11      12      13      14      15

       16      17      18      19      20
```

```
   21     22     23     24     25
```

```
>> A51(:,3)
```

```
ans =
```

```
     3
      8
    13
    18
    23
```

```
>> A51(3,:)
```

```
ans =
```

```
    11     12     13     14     15
```

```
>> A51(3,3)
```

```
ans =
```

```
    13
```

```
>> A51(:,:)
```

```
ans =
```

```
     1      2      3      4      5
     6      7      8      9     10
```

```
    11      12      13      14      15

    16      17      18      19      20

    21      22      23      24      25


>> Atest=A51(:,4)


Atest =


     4

     9

    14

    19

    24


>> Ctest=A51(5,2)


Ctest =


    22


>> A51(3,3)=42


A51 =


     1       2       3       4       5

     6       7       8       9      10

    11      12      42      14      15

    16      17      18      19      20

    21      22      23      24      25
```

```
>> A41

A41 =

     1      2      3      4
     2      4      7     11
     3      7     14     25
     4     11     25     50

>> B41

B41 =

     5
    10
    15
    20

>> B42

B42 =

     5     10     15     20

>> A41*B41

ans =

        150
        375
```

```
      795

      1505
```

```
>> A41*B42
```

Error using *

Incorrect dimensions for matrix multiplication. Check that the number of columns in the

first matrix matches the number of rows in the second matrix. To operate on each element of

the matrix individually, use TIMES (.*) for elementwise multiplication.

Related documentation

```
>> matrices
```

```
>> A41*B41
```

Error using *

Incorrect dimensions for matrix multiplication. Check that the number of columns in the

first matrix matches the number of rows in the second matrix. To operate on each element of

the matrix individually, use TIMES (.*) for elementwise multiplication.

Related documentation

```
>> A41*b42
```

Unrecognized function or variable 'b42'.

Did you mean:

```
>> A41*B42A41*B42
```

Unrecognized function or variable 'B42A41'.

```
>> A41*B42


ans =


          150
          375
          795
         1505


>> %I may or may not have accidently switched B41 and B42 initaily...
>> A51(3,3)=42


A51 =


     1     2     3     4     5
     6     7     8     9    10
    11    12    42    14    15
    16    17    18    19    20
    21    22    23    24    25


>> %everything should be back to how its supposed to be now...
>> A41*B41.'


ans =


          150
          375
          795
         1505
```

```
>> A21^-1

ans =

    8.0000   -3.0000
   -5.0000    2.0000

>> inv(A21)

ans =

    8.0000   -3.0000
   -5.0000    2.0000

>> invA(22)
Unrecognized function or variable 'invA'.

Did you mean:
>> inv(A22)
Warning: Matrix is singular to working precision.

ans =

    Inf    Inf
    Inf    Inf

>> det(A21)
```

```
ans =


    1.0000


>> det(A22)


ans =


     0


>> inv(A41)


ans =


   -6.0000    9.0000   -5.0000    1.0000
    9.0000   -1.0000   -5.0000    2.0000
   -5.0000   -5.0000    9.0000   -3.0000
    1.0000    2.0000   -3.0000    1.0000


>> det(A41)


ans =


   -1.0000


>>
```

Part 1 B

```
>> matrices
>> A21*eye(2)
```

```
ans =


     2      3
     5      8


>> A22*eye(2)


ans =


     6      3
     8      4


>> A41*eye(4)


ans =


     1      2      3      4
     2      4      7     11
     3      7     14     25
     4     11     25     50


>> A51*eye(5)


ans =


     1      2      3      4      5
     6      7      8      9     10
    11     12     13     14     15
    16     17     18     19     20
```

```
    21    22    23    24    25
```

Part 2

```
>> A23\B23


ans =


    1.5263
    1.1579


>> A23^-1*B23


ans =


    1.5263
    1.1579


>> inv(A23)*B23


ans =


    1.5263
    1.1579


>> AA23 = [A23 B23]


AA23 =


     2     6    10
```

```
    -5      4     -3
```

```
>> rref(AA23)
```

```
ans =
```

```
    1.0000         0     1.5263
         0    1.0000     1.1579
```

```
>> rref([A31 B31])
```

```
ans =
```

```
    1.0000         0         0     2.7500
         0    1.0000         0     4.2500
         0         0    1.0000     9.2500
```

```
>> rref([A32 B32)]
 rref([A32 B32)]

          ↑
Invalid expression. When calling a function or indexing a variable,
use parentheses.

Otherwise, check for mismatched delimiters.
```

```
Did you mean:
>> rref([A32 B32])
```

```
ans =
```

```
    1      0     -1      2
    0      1      2     -1
```

```
     0      0      0      0


>> rref([A33 B33])


ans =


     1      0     -1      0
     0      1      2      0
     0      0      0      1


>>
A41*inv(A41)


ans =


    1.0000   -0.0000         0         0
    0.0000    1.0000         0         0
         0   -0.0000    1.0000   -0.0000
         0   -0.0000         0    1.0000


>> A21*inv(A21)


ans =


     1      0
     0      1


>>
```

Post-Lab

```
>> eye(1)
```

```
ans =


     1


>> eye(2)


ans =


     1     0
     0     1


>> eye(3)


ans =


     1     0     0
     0     1     0
     0     0     1


>> eye(4)


ans =


     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1


>> eye(5)
```

```
ans =

     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1

>>
```

1. Part a and b are in 1B

2.

```
det(B41)

Error using det

Matrix must be square.


>> det(B42)

Error using det

Matrix must be square.


>> inv(B41)

Error using inv

Matrix must be square.


>> inv(B42)

Error using inv

Matrix must be square.
```

There was a error because the matrices are not square and to take a determinate or an inverse the matrix needs to be square.

3.

$$3.$$
a.
$$\begin{cases} 2x+6y=10 \\ -5x+4y=-3 \end{cases}$$

$$\frac{2x=10-6y}{2}$$
$$x=5-3y$$

$$-5(5-3y)+4y=-3$$
$$-25+15y+4y=-3 \qquad -25+19y=-3$$
$$\qquad\qquad\qquad\qquad\qquad -25 \qquad\quad +25$$

$$x=5-3\left(\tfrac{22}{19}\right) \qquad\quad \frac{19y}{19}=\frac{22}{19} \qquad y=\frac{22}{19}$$

b.
$$\begin{cases} x+2y-3z=26 \\ -x+8y+2z=24 \\ 13x+2y+7z=-14 \end{cases}$$

$$(x+2y-3z)+(-x+8y+2z)=26+24$$
$$10y-z=50$$

$$13(x+2y-3z)-(13x+2y+7z)=13\cdot26-(-14)$$
$$24y-32z=352$$

$$13$$

$$z=50-10y$$

$$24y-32(50-10y)=352 \qquad 24y-1600+320y=352$$

$$z=50-10(5.67) \qquad 344y-1600=352 \qquad \frac{344y}{344}=\frac{1952}{344} \approx 5.67=y$$

$$z=50-56.7 \quad z=-6.7$$

$$x+2(5.67)-3(-6.7)=26$$
$$x+31.44=26 \quad x=-5.44$$

4.

```
det(A32)


ans =


    1.3323e-15


>> det(A33)
```

```
ans =


    6.6613e-16


>> inv(A32)

Warning: Matrix is close to singular or badly scaled. Results may be
inaccurate. RCOND =

1.321694e-18.



ans =


    1.0e+16 *


   -0.2252     0.9007    -0.4504

    0.4504    -1.8014     0.9007

   -0.2252     0.9007    -0.4504


>> inv(A33)

Warning: Matrix is close to singular or badly scaled. Results may be
inaccurate. RCOND =

1.541976e-18.



ans =


    1.0e+16 *


   -0.4504     0.9007    -0.4504

    0.9007    -1.8014     0.9007

   -0.4504     0.9007    -0.4504
```

>>

The determinants are close to zero or are zero and that would give a divide by zero error when calculating the inverse. These answers are not reasonable. I cannot tell the difference between the two matrices with these calculations.

5.

```
>> A1 = [1 1 1;2 -1 1;1 2 -1]


A1 =


     1      1      1
     2     -1      1
     1      2     -1


>> A2 = [6;3;4]


A2 =


     6
     3
     4


>> rref([A1 A2])


ans =


    1.0000         0         0    1.5714
         0    1.0000         0    2.2857
         0         0    1.0000    2.1429


>> B1 = [1 2 -3;2 -4 6;3 6 -9]
```

```
B1 =

     1      2     -3
     2     -4      6
     3      6     -9


>> B2 = [7;-14;15]


B2 =

      7
    -14
     15


>> rref([B1 B2])


ans =

    1.0000         0         0         0
         0    1.0000   -1.5000         0
         0         0         0    1.0000


>> C1 = [1 1 1;2 2 2;3 3 3]


C1 =

     1      1      1
     2      2      2
     3      3      3
```

```
>> C2 = [6; 12; 18]


C2 =


    6

   12

   18


>> rref([C1 C2])


ans =


    1      1      1      6

    0      0      0      0

    0      0      0      0


>> D1 = [2 -3 4;5 6 -7;8 -9 10]


D1 =


    2     -3      4

    5      6     -7

    8     -9     10


>> D2 = [20;-14;30]


D2 =


   20
```

```
    -14

     30


>> rref([D1 D2])


ans =


    1.0000         0         0    1.1000
         0    1.0000         0   15.5333
         0         0    1.0000   16.1000



>> E1 = [4 -5 6;7 8 -9;10 -11 12]


E1 =


     4     -5      6
     7      8     -9
    10    -11     12


>> E2 = [14;-16;18]


E2 =


    14
   -16
    18


>> rref([E1 E2])
```

```
ans =

    1.0000         0         0         0
         0    1.0000         0   10.0000
         0         0    1.0000   10.6667


    >>
```

c has infinite solutions and B has no solutions. I learned a much faster way to solve systems of equations and more about linear algebra.

6.I was just about able to finish during lab time with maybe 20 min outside of the lab.

7. About an hour.