MAT 343 LAB 2 - Jordan Ledbetter

Question 1

(a)

Create the matrix A and the vectors z and b for the given value of n

```
n = 1100;
A = floor(15*rand(n));
z = ones(n,1);
b = A*z;
```

(i)

Measure the compuational time for the two methods

```
tic, x = A\b; toc
```

```
Elapsed time is 0.077804 seconds.
```

```
tic, y = inv(A)*b; toc
```

```
Elapsed time is 0.183473 seconds.
```

Which method is faster?

Method x is faster than method y.

(ii)

Compare the accuracy of the two methods

```
sum(abs(x-z))
```

```
ans = 3.4064e-10
```

```
sum(abs(y-z))
```

```
ans = 1.2902e-09
```

Which method is more accurate?

Solution x is more accurate to z than solution y.

(b1)

Create the matrix A and the vectors z and b for the given value of n

```
n = 2200
```

```
n = 2200
```

```
A = floor(15*rand(n));
z = ones(n,1);
b = A*z;
```

(i)

Measure the compuational time for the two methods

```
tic, x = A\b; toc
```

```
Elapsed time is 0.157019 seconds.
```

```
tic, y = inv(A)*b; toc
```

```
Elapsed time is 0.283354 seconds.
```

Which method is faster?

When n = 2200, method x is faster than method y.

(ii)

Compare the accuracy of the two methods

```
sum(abs(x-z))
```

```
ans = 2.0608e-09
```

```
sum(abs(y-z))
```

```
ans = 1.3785e-08
```

Which method is more accurate?

When n = 2200, the \ method is more accuracte than the inverse method.

(b2)

Create the matrix A and the vectors z and b for the given value of n

```
n = 4400;
A = floor(15*rand(n));
z = ones(n,1);
b = A*z;
```

(i)

Measure the compuational time for the two methods

```
tic, x = A\b; toc
```

```
Elapsed time is 0.768690 seconds.
```

```
tic, y = inv(A)*b; toc
```

```
Elapsed time is 1.880423 seconds.
```

Which method is faster?

When n = 4400, method x is faster than method y.

(ii)

Compare the accuracy of the two methods

```
sum(abs(x-z))
```

```
ans = 1.6462e-08
```

```
sum(abs(y-z))
```

```
ans = 7.2664e-08
```

Which method is more accurate?

When n = 4400, the inverse method is more accurate than the \ method.

(c)

Explain why the exact solution of the system Ax= b is the vector z.

Explanation: Vector Z is a n x 1 matrix with all entries equal to 1. Since Ax=b is the same as A^-1*b, we can see that all entries equal one.

Question 2

Generate the matrices B and A and the vectors b and z using the given commands

```
n = 40;
B = eye(n) - triu(ones(n),1);
A = B'*B;
z = ones(n,1);
b = A*z;
```

Compute the solution using the mldivide operator and using the inverse

```
x = A\b;
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
6.670811e-27.
```

```
y = inv(A)*b;
```

```
Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND =
6.670811e-27.
```

Compare the accuracy of the two methods

```
sum(abs(x-z))
```

```
ans = 0
```

```
sum(abs(y-z))
```

```
ans = 2.9457e+09
```

Which method produces the more accurate solution?

The \ method produced a more accurate results than the inverse method.

Question 3

Generate the matrix A and the vector b using the given commands

```
A = floor(10*rand(6));
b = floor(20*rand(6,1))-10;
```

(a)

```
x = A\b
```

```
x = 6×1
    1.2229
   -1.1671
    2.2230
   -0.5122
   -1.5399
   -0.6164
```

(b)

```
U = rref([A,b])
```

```
U = 6×7
    1.0000         0         0         0         0         0    1.2229
         0    1.0000         0         0         0         0   -1.1671
         0         0    1.0000         0         0         0    2.2230
         0         0         0    1.0000         0         0   -0.5122
         0         0         0         0    1.0000         0   -1.5399
         0         0         0         0         0    1.0000   -0.6164
```

(c)

```
U(:,7)-x
```

```
ans = 6×1
10^-5 ×
    0.9570
    0.0317
   -0.0157
    0.1218
    0.7961
    0.7485
```

Are the answers computed with the rref and with the "\" the same?

Answer: Yes, the final column of the reduced row echelon form produces the same results as the "\" method.

(d)

```
A(:,6) = 7*A(:,5)+4*A(:,1);
rref([A,b])
```

```
ans = 6×7
     1     0     0     0     0     4     0
     0     1     0     0     0     0     0
     0     0     1     0     0     0     0
     0     0     0     1     0     0     0
     0     0     0     0     1     7     0
```

```
     0     0     0     0     0     0     1
```

How many solutions does the system have?

Answer: The system has no solutions since the last row is filled with all zeros and a singular one.

(e)

Create the vectors y and c:

```
y = floor(20*rand(6,1)) - 10;
c = A*y;
```

Why is the system Ax = c guaranteed to be consistent?

Answer: The system Ax=c is guaranteed to be consistent because c is the product of two random vectors, and the system would only be inconsistent if the product of c was a result of elements with the same value.

(f)

```
U = rref([A,c])
```

```
U = 6×7
     1     0     0     0     0     4   -31
     0     1     0     0     0     0     0
     0     0     1     0     0     0     0
     0     0     0     1     0     0     3
     0     0     0     0     1     7   -40
     0     0     0     0     0     0     0
```

How many solutions does the system have?

Answer: The system has infinitely many solutions since the last row contains all values of zero.

Question 4

```
 % print the function file
type myrowproduct.m
```

```
function [y]=myrowproduct(A,x)
[m,n]=size(A);
[p,q]=size(x);
if n==p
    for i=1:m
        y(i,1)=A(i,:)*x;
    end
else
    display ('dimensions do not match')
    y = []
end
end
```

*Generate the random matrix A and vector x and test your function by comparing the output with the product A*x*

```
A = rand(5,4)
```

```
A = 5×4
    0.0517    0.8492    0.6009    0.5088
```

```
    0.7320    0.5076    0.6290    0.3459
    0.7763    0.6132    0.4487    0.8857
    0.1376    0.9771    0.0914    0.5215
    0.1621    0.5329    0.9491    0.9960
```

```
x = rand(4,1)
```

```
x = 4×1
    0.0965
    0.9974
    0.3304
    0.4620
```

```
A*x
```

```
ans = 5×1
    1.2856
    0.9445
    1.2440
    1.2590
    1.3209
```

```
y = myrowproduct(A,x)
```

```
y = 5×1
    1.2856
    0.9445
    1.2440
    1.2590
    1.3209
```

```
A = rand(4,2)
```

```
A = 4×2
    0.6070    0.5185
    0.8604    0.4064
    0.8665    0.8688
    0.1131    0.9215
```

```
x = rand(2,1)
```

```
x = 2×1
    0.7384
    0.8132
```

```
A*x
```

```
ans = 4×1
    0.8698
    0.9658
    1.3463
    0.8328
```

```
y = myrowproduct(A,x)
```

```
y = 4×1
    0.8698
    0.9658
    1.3463
    0.8328
```

```
A = rand(4,2)
```

```
A = 4×2
    0.3213    0.6650
    0.3939    0.1505
    0.7934    0.6660
    0.5632    0.5313
```

```
x = rand(1,2)
```

```
x = 1×2
    0.0858    0.1173
```

```
%A*x
y = myrowproduct(A,x)
```

```
dimensions do not match

y =

     []


y =

     []
```

Question 5

(a)

```
type columnproduct.m   % print the function file
```

```
function C = columnproduct(A,B)
M = size(A);
N = size(B);
if(M(2)~=N(1))
    fprintf('Matrix dimensions do not match');
    return;
end
C = [];
for k = 1:B
    C(:,k) = A*B(:,k);
end
end
```

*Generate the random matrices A and B and test your function by comparing the output with the product A*B*

```
A = rand(4,4)
```

```
A = 4×4
    0.7200    0.9207    0.4004    0.4319
    0.4835    0.1658    0.9035    0.2826
    0.3267    0.4274    0.0361    0.2054
    0.5037    0.2642    0.9728    0.0730
```

```
B = rand(4,2)
```

```
B = 4×2
    0.7275    0.8056
    0.3428    0.0067
    0.5586    0.6143
    0.3883    0.5627
```

```
A*B
```

```
ans = 4×2
    1.2309    1.0753
    1.0230    1.1047
    0.4841    0.4038
    1.0288    1.0462
```

```
y = columnproduct(A,B)
```

```
y =

     []
```

```
A = rand(2,6)
```

```
A = 2×6
    0.3501    0.6576    0.4478    0.1369    0.1308    0.5321
    0.7437    0.4511    0.2528    0.3579    0.4895    0.6729
```

```
B = rand(6,5)
```

```
B = 6×5
    0.9450    0.7883    0.1755    0.5106    0.1646
    0.0630    0.5437    0.2280    0.2318    0.5984
    0.4629    0.0235    0.7480    0.8938    0.7930
    0.2549    0.4431    0.5460    0.4591    0.5291
    0.8696    0.2840    0.2396    0.3704    0.1669
    0.1474    0.9005    0.7218    0.8974    0.1046
```

```
A*B
```

```
ans = 2×5
    0.8066    1.2210    1.0365    1.3202    0.9561
    1.4642    1.7410    1.2208    1.6597    0.9342
```

```
y = columnproduct(A,B)
```

```
y =

     []
```

```
A = rand(2,6)
```

```
A = 2×6
    0.5824    0.0560    0.0183    0.8760    0.3259    0.4329
    0.9983    0.2225    0.4019    0.5447    0.1840    0.2911
```

```
B = rand(5,6)
```

```
B = 5×6
    0.4993    0.2012    0.7868    0.0594    0.9576    0.4717
    0.7458    0.3787    0.9657    0.2264    0.9599    0.9572
    0.8110    0.8443    0.4631    0.4129    0.7559    0.0581
    0.0533    0.8440    0.7368    0.0042    0.4585    0.1195
```

```
    0.9435    0.3187    0.5383    0.9599    0.9826    0.4797
```

```
%A*B
%y = columnproduct(A,B)
```

(b)

```
type rowproduct.m
```

```
function C = rowproduct(A,B)
M = size(A);
N = size(B);
if(M(2)~=N(1))
    fprintf('Matrix dimensions do not match');
    return;
end
C = [];
for k = 1:M(1)
    C = [C;A(k,:)*B];
end
end
```

*Generate the random matrices A and B and test your function by comparing the output with the product A*B*

```
A = rand(4,4)
```

```
A = 4×4
    0.1838    0.1405    0.2981    0.2355
    0.8841    0.8570    0.6807    0.8465
    0.0419    0.9277    0.6883    0.7621
    0.4686    0.2271    0.6260    0.4659
```

```
B = rand(4,2)
```

```
B = 4×2
    0.4092    0.2504
    0.6007    0.9204
    0.2667    0.4650
    0.7620    0.4095
```

```
A*B
```

```
ans = 4×2
    0.4185    0.4103
    1.7031    1.6733
    1.3387    1.4964
    0.8502    0.8082
```

```
y = rowproduct(A,B)
```

```
y = 4×2
    0.4185    0.4103
    1.7031    1.6733
    1.3387    1.4964
    0.8502    0.8082
```

```
A = rand(2,6)
```

```
A = 2×6
```

```
    0.6823    0.5828    0.3451    0.9344    0.2769    0.0998
    0.6566    0.4030    0.3583    0.6220    0.0289    0.5143
```

```
B = rand(6,5)
```

```
B = 6×5
    0.6814    0.9498    0.3202    0.6646    0.0720
    0.8426    0.5189    0.7454    0.0961    0.1021
    0.9554    0.4385    0.1141    0.1231    0.0983
    0.8675    0.8420    0.7591    0.1287    0.1193
    0.6494    0.7811    0.5335    0.0447    0.1309
    0.4731    0.6831    0.5982    0.1850    0.6326
```

```
A*B
```

```
ans = 2×5
    2.3233    2.1730    1.6090    0.7030    0.3534
    1.9309    1.8874    1.3467    0.6957    0.5270
```

```
y = rowproduct(A,B)
```

```
y = 2×5
    2.3233    2.1730    1.6090    0.7030    0.3534
    1.9309    1.8874    1.3467    0.6957    0.5270
```

```
A = rand(2,6)
```

```
A = 2×6
    0.4604    0.4502    0.2439    0.0142    0.2791    0.7968
    0.0875    0.6356    0.1054    0.0581    0.1515    0.5674
```

```
B = rand(5,6)
```

```
B = 5×6
    0.7537    0.4061    0.9273    0.4442    0.8463    0.7768
    0.7788    0.4141    0.2934    0.1996    0.1955    0.6317
    0.8071    0.4761    0.7394    0.0485    0.0015    0.7072
    0.5200    0.6981    0.1106    0.3327    0.9494    0.4638
    0.2730    0.7059    0.4156    0.8826    0.2186    0.7340
```

```
%A*B
y = rowproduct(A,B)
```

```
Matrix dimensions do not match
Output argument "C" (and possibly others) not assigned a
value in the execution with "rowproduct" function.
```