MTH 420/520 Homework 5. **Yesh Godse**
Date: May 22, 2019
Prof. M. Peszynska

MTH 420 students turn in 1-2 or more for extra credit. MTH 520 students turn in 1-2-3.
Show enough work and enough of `MATLAB code` to demonstrate this is your own hard work, but be concise.
Sloppy incomplete work, or lengthy algebraic calculations, or core dump of MATLAB output with errors will not receive much credit.

Problem 1. Recall Pbm Ex.II.9 from class notes. The code below illustrates the kernel function `gfun` and a function called `gravity` (you can save it as an M-file).

```
function [K] = gravity(n,D )
    K = zeros(n,n);
    h = 1/n;
    gfun =@(x)(D./(sqrt(D^2+x.^2)).^3;
    distance=linspace(0,1);plot(distance,gfun(distance)); pause;
    for j=1:n
        for m=1:n
            K(j,k) = gfun((j-m)/n);
        end
    end
    K = h*K;
end
```

Discuss the properties of the gravity function `gfun` and the properties of matrix K obtained by the `gravity`. For example, choose D=1e-1,1e0,1e1, and $n = 10, 100$. Specifically, report on (A) `cond(K)` depending on n and D. (B) Find the first singular value of K less than 10^{-8}.
(**Extra:**) consider the case when K is not square, and when you have m observations, and want to determine the treasure in n location. Discuss what you expect from this case.

Solution 1. The following table shows the value of `cond(K)` for different values of n and D. The value returned by `cond(K)` represents the sensitivity of the inverse problem. A larger value indicates a greater sensitivity. As n and D increase, the sensitivity of the inverse problem increases as well. This makes sense, as a greater depth D and smaller step size $h = \frac{1}{n}$ would make finding the source f very sensitive to the position of the receiver.

The table also notes the first singular value for each combination of n and D that is less than 10^{-8}. As n and D increase, the singular values of K generally decrease, and the first such value less than 10^{-8} comes earlier on. This is indicative of a weaker signal for where the true f is.

n	D	`cond(K)`	first singular value less than 10^{-8}
10	0.1	4.318	none
	1.0	1.475*10^{9}	10th singular value (out of 10)
	10	3.587*10^{17}	4th singular value (out of 10)
100	0.1	2.739*10^{12}	77th singular value (out of 100)
	1.0	9.311*10^{18}	10th singular value (out of 100)
	10	1.459*10^{20}	4th singular value (out of 100)

Problem 2. The following code shows what happens if you apply the code to uncover a "treasure" f buried in three pieces underground. The code computes the signal g in the receiver. We try next to solve the inverse problem starting from g to f when g is perturbed. In your experiments focus on $n = 10$ and the same cases of D as in Problem 1.

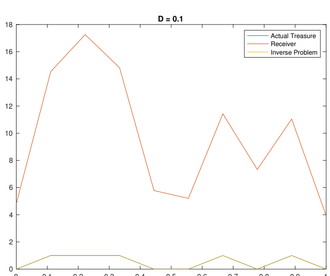
```
%%
n=10;  D=.1;  K = gravity(n,D);
mytreasure = [0,1,1,1,0,0,1,0,1,0]';
%% (1 means present, 0 means absent).
f = mytreasure; g = K*f;
t=linspace(0,1,length(mytreasure))';
plot(t,f,t,g); %% or plot in a more fancy way
pause;
%% now solve the inverse problem
newf = K \g;
norm(newf-f)
%% perturb rhs and try to find the treasure now
rand_mag=0.5;
gp = g + rand_mag*rand(n,1); fp=K\gp;
```

(A) (Start with $D = 0.1$ and $n = 10$). Does the shape of g resemble the "shape" of f? Clearly K smooths the rough f. Too much? Connect to A or B from Problem 1.
(B) Is `newf` the same as `f`? How is this connected to the cond(K)? Does the result fp resemble f? To see the effect, experiment with different rand_mag. **Discuss** the connection to Problem 1.
Extra credit: Experiment with different n and m. When changing n, you must change "mytreasure"!

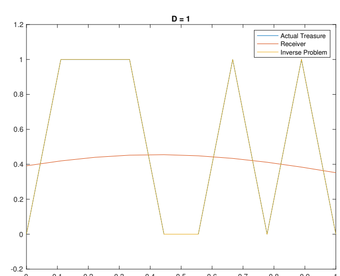
Solution 2. The following code computes and visualizes the forward problem of finding g from K and f, the inverse problem of finding f from K and g, and seeing the effects of perturbation on the inverse problem. This is done for all combinations of $n = 10$ and $D = \{0.1, 1.0, 10.0\}$

```
D = [0.1000  1.0000  10.0000]; n=10;
mytreasure = [0,1,1,1,0,0,1,0,1,0]';
for i=1:3
    K = gravity(n,D(i)); f=mytreasure; g=K*f;
    t=linspace(0,1,length(mytreasure))';
    % Now solve the inverse problem
    newf = K \ g; norm(newf-f)
    % Perturb rhs and try to find the treasure now
    rand_mag=0.5; gp = g + rand_mag*rand(n,1);
    fp = K \ gp;
    % Plotting
    figure(); plot(t,f,t,g,t,newf); %% or plot in a more fancy way
    legend("Actual Treasure","Receiver","Inverse Problem");
    figure(); plot(t,f, t,fp);
    legend("Actual Treasure","Perturbed Inverse Problem");
end
```

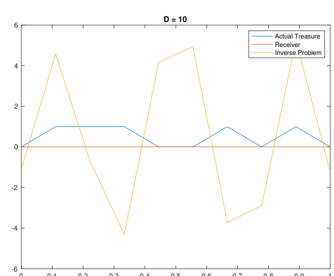
For $n = 10$ and $D = 0.1$, the shape of g does resemble the shape of f. When $D \geq 1.0$, K smooths the rough f too much. Note that the f computed from the inverse problem, shown in yellow, is identical to the original f for $D = 0.1$ and 1.0. As a result, they overlap in figures 1A and 1B. When $D = 10$, the f computed from the inverse problem has no resemblance to the original f. The different results for different values of D reflect the increasing sensitivity of the problem as D increases. This relationship was revealed in Problem 1.



(A) D=0.1



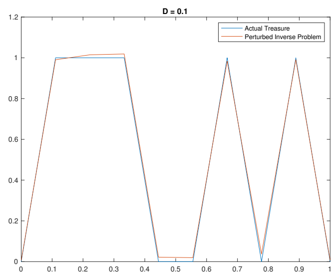
(B) D=1.0



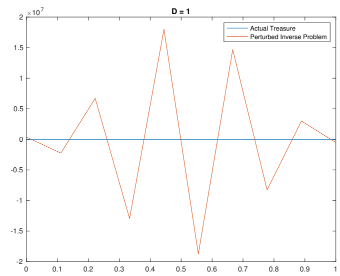
(C) D=10

FIGURE 1. Plots of f, g, and f computed from inverse problem

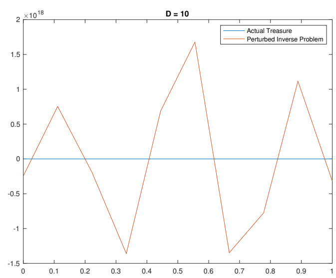
The figure below shows the perturbed inverse problem compared to f for various values of D and `rand_mag = 0.5`. For $D = 0.1$, the perturbation does not lead to a significant distinction between the actual f and fp computed from the (perturbed) inverse problem. For other values of D, the perturbation doesn't have any noticeable effect, as these values of D lead to an oversensitive inverse problem.



(A) D=0.1



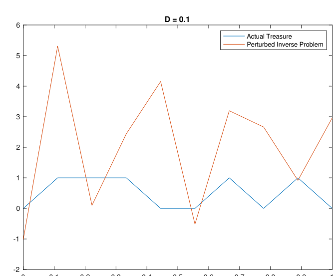
(B) D=1.0



(C) D=10

FIGURE 2. Plots of f and fp computed from inverse problem, `rand_mag = 0.5`

If we change the value of `rand_mag` to 50, the fp doesn't resemble f even at $D = 0.1$. Between these values, the K corresponding to $D = 0.1$ is robust enough to perturbations (not oversensitive) to still return a close approximation of f despite the noise.



(A) D=0.1

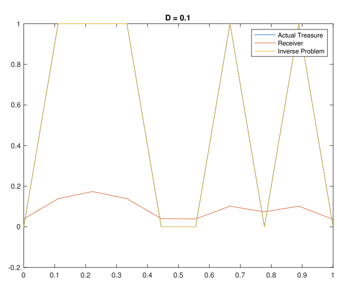
FIGURE 3. Plots of f and fp computed from inverse problem, `rand_mag = 50`

Problem 3. Follow the example above for the bar-code reading problem. Attempt to answer the same questions as in Problems 1-2.

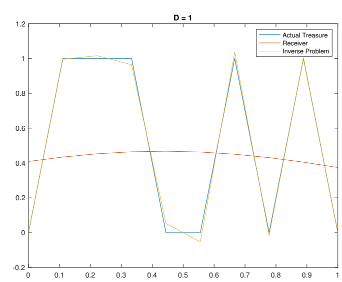
Solution 3. The following MATLAB code shows the `gravity` function modified for the barcode problem in class notes. The notable difference is the definition of `gfun`. In this function, the parameter `D` corresponds to the accuracy of the barcode scanner σ.

```
function [K] = barcode(n,D )
    K = zeros(n,n);
    h = 1/n;
    gfun =@(x)(exp(-(x.^2)/(D^2)));
    distance=linspace(0,1);
    plot(distance,gfun(distance));
    for j=1:n
        for m=1:n
            K(j,m) = gfun((j-m)/n);
        end
    end
    K = h*K;
end
```

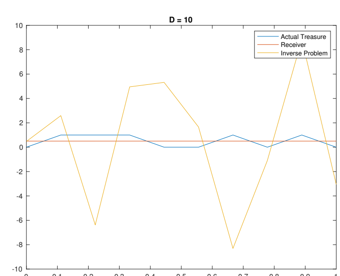
The following plots show f, g, and f computed via the inverse problem for $\sigma = \{0.1, 1.0, 10.0\}$. In all cases, $n = 10$. The results are analogous to those for the underwater treasure problem, with $D = 0.1$ being the only case where K does not over-smooth f.



(A) D=0.1



(B) D=1.0



(C) D=10

FIGURE 4. Plots of f, g, and f computed from inverse problem