Inverse Problem Midterm

Shuowen Wei

1.

Nonlinear inverse problem is usually like

$$y = f(x)$$

where f is a nonlinear operator and cannot be represented as a combination of linear mappings of parameter x to the observed data y. The structure of f is very important in solving for x, different properties of f lead to different ways of solving this problem.

Here, we give an example of one-diminsion nonlinear problem and discuss how to solve it. Consider problem as follows:

$$f(x) = x^3 - x - 1 = 0$$

Solve it for all the positive roots. This is obviously nonlinear inverse problem because the highest order of x is 3.

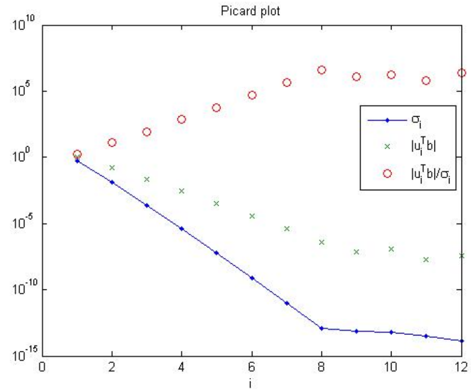
By Descartes' rule of signs, we know that this function has only one positive root because there is only one variation in sign of the sequence of coefficients. By letting its first derivative $f'(x) = 3x^2 - 1 > 0$ we know that this function is monotone increasing on $[\frac{\sqrt{3}}{3}, +\infty)$, since $f(1) = -1 < 0$ and $f(2) = 5 > 0$ we know that the root lie on [1,2].

Rewrite the equation as $g(x) = \sqrt[3]{x+1}$, then it's easy to verify $g: [1,2] \rightarrow [1,2]$. Since $\forall x \epsilon [1,2], |g'(x)| < 1$, thus g is a contraction map on [1,2]. Then the Contraction Mapping Theorem, there is a unique fixed point x on [1,2] such that $g(x) = x$, i.e. the root of $f(x) = 0$.

Then by the iteration method, let $x_{k+1} = \sqrt[3]{x_k + 1}$ and we can just set $x_o = 1.5$ and run enough times, say 100, in MATLAB and we will finally get the fixed point 1.3247, which is the solution.

2.

Set *n=12*, we use **ursell** to generate *A*, *b* and then compute the SVD of *A*. Aftering using the function **picard** to analyse the problem, we got the graph below:

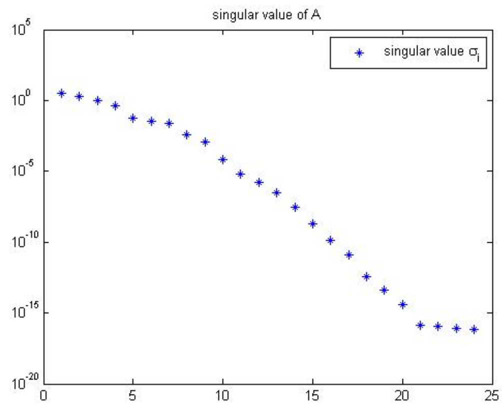


From the graph we can see that the Fourier coefficients (the green stars) decay slower than the singular values (the blue line-dots) of *A* greater than the machine precision level, which violates the discrete picard condition. That's because the integral equation has no square integrable solution. [1]

3.

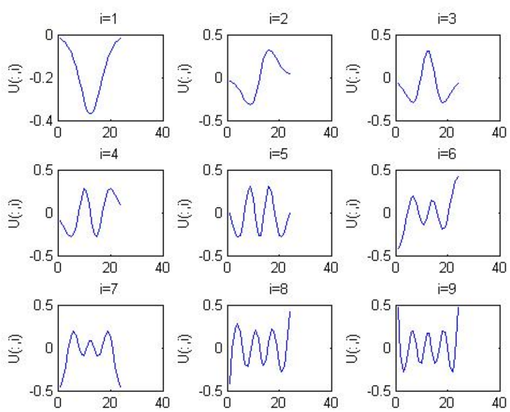
3.1

Set *n=24*, by using function **shaw** we get the matrix *A*, the right-hand b and the exact solution *x*. After computing the SVD of *A*, we plot its singular values out:

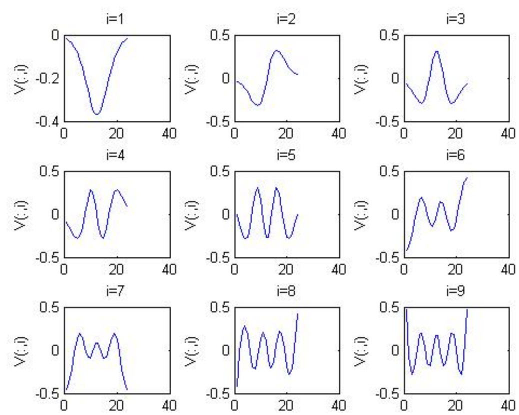


3.2

Then we can see that the front 10 singular values relatively big while the rest about a dozen of singular values are very tiny, thus only the 10 front left and right singular values capture most of the matrix A's information, while the rest don't carry reliable information about the singular functions. Thus for the convenience of plotting graph, we choose to plot out the front 9 singular vectors, as follows: [2]



Left singular vectors U

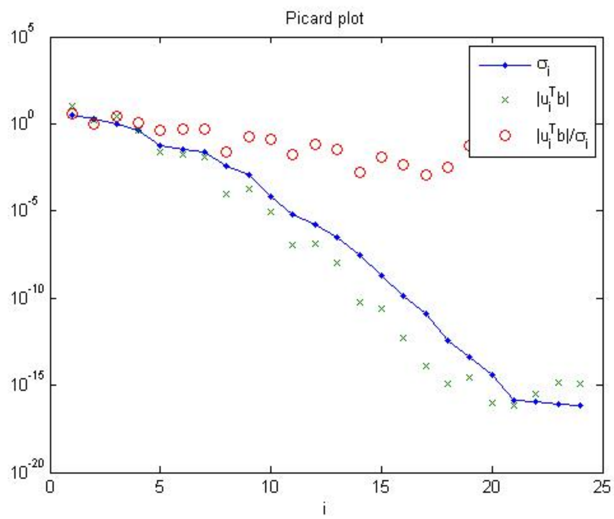


Right singular vectors V

By inspecting the number of sign changes, we can easily see that the singular vectors have more oscillations as the index i increases, i.e. the corresponding singular value decrease. In this problem, the number of sign changes in the ith singular vectors is $i - 1$.

3.3

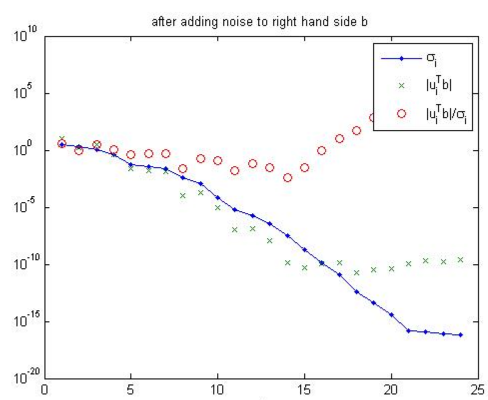
By using the **picard** function, we get:



The picard plot shows that the coefficients (the green stars) decay faster than the singular values (the blue line-dots) until $i > 21$ at a plateau determined by the machine precision, and these tiny coefficients are less than the machine preecision times $\|b\|_2$. The solution coefficients $\mu_i^T b/\sigma_i$ decay for $i \leq 17$ but for $i > 17$ they start to increase again due to the inaccurate values of $\mu_i^T b$. So obviously, the picard condition is satisfied. [3]

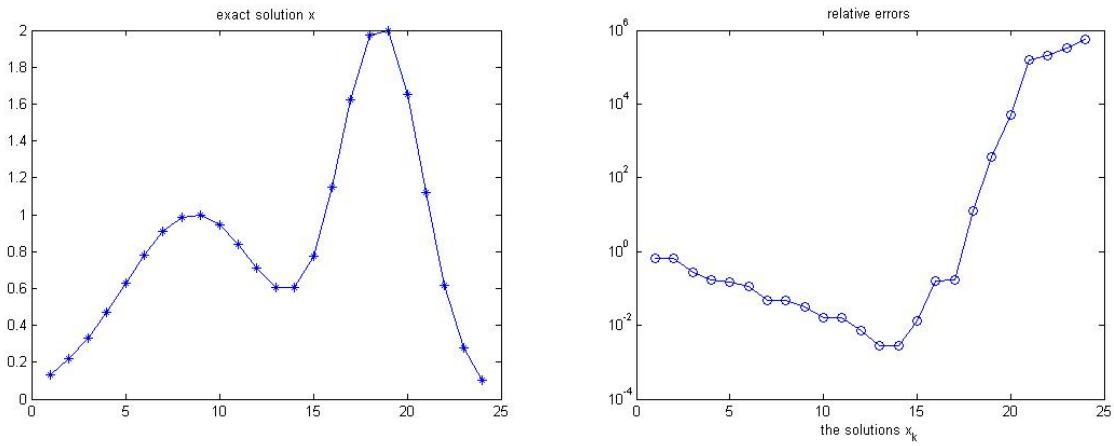
3.4

After adding a small amount of noise e to the right-hand side b, we use picard function to check it and find that the Fourier coefficients corresponding to the small singular values begin decay slower than the small singular values, showed in the right:

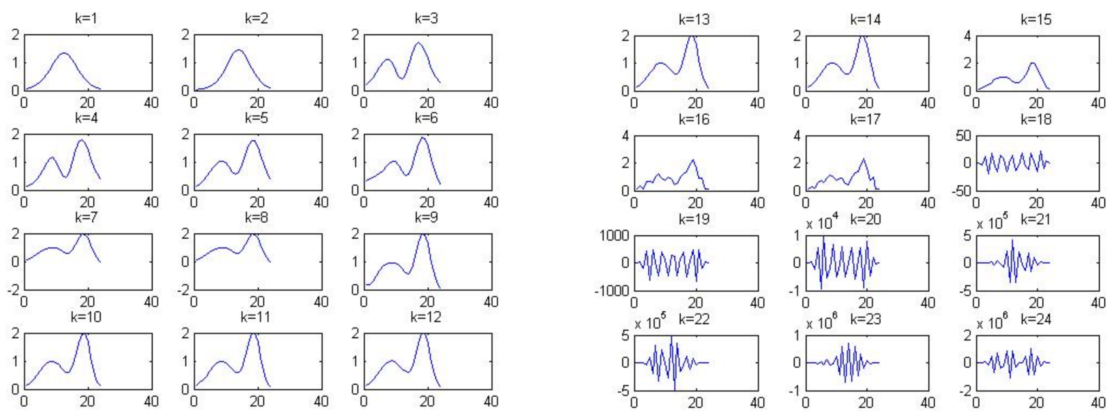


3.5

First of all, we plot out the exact solution x showed in the first graph. Then we use **tsvd** to compute every xk where k from 1 to 24 and then compute each x_k's relative error with respect to the exact solution x, showed in the second graph.



By analysis the relative solutions of each x_k, we can see that as the index k increase, the x_k get closer and closer to the exact solution x until $k = 13$, after when the relative errors jump very largely because of the tiny and inaccurate values of the singular values. We also show all the x_k below in the two graphs:

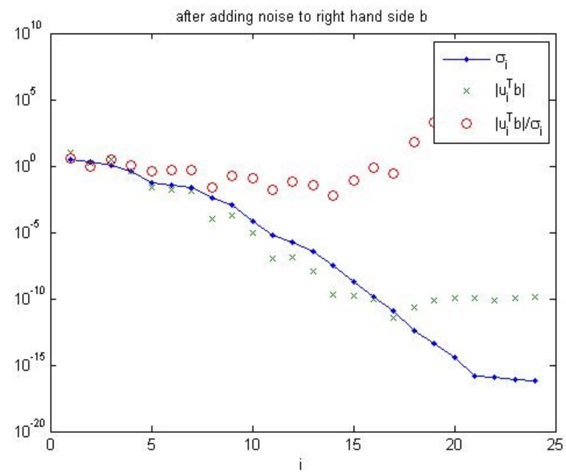


3.6

With a larger level noise we repeat 4 and 5 as follows:

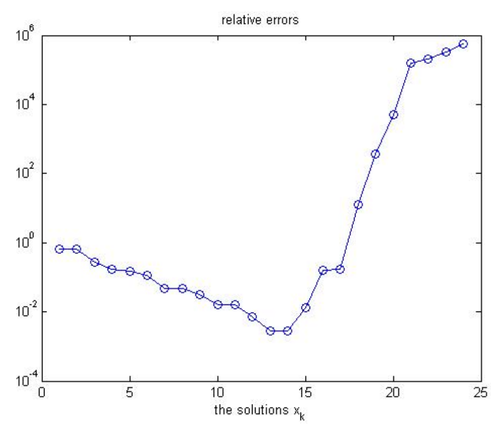
For 4:

When the noise is larger, the graph seems very likely to the one with a smaller noise. The Fourier coefficients corresponding to the small singular values begin decay slower than the small singular values at about the same index i, showed in the right.

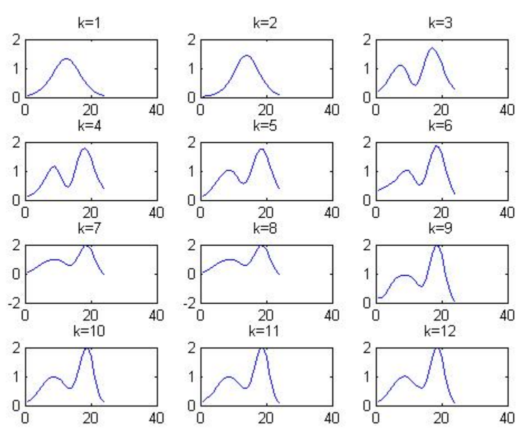


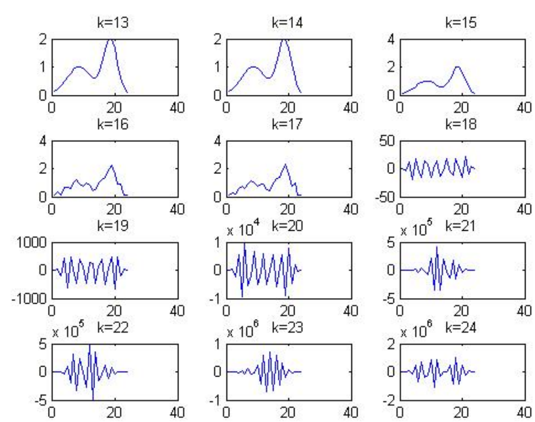
For 5:

Since we already showed the exact solution above, the new relative errors are showed in the right, it still gets its best approximation solution at $k = 13$:



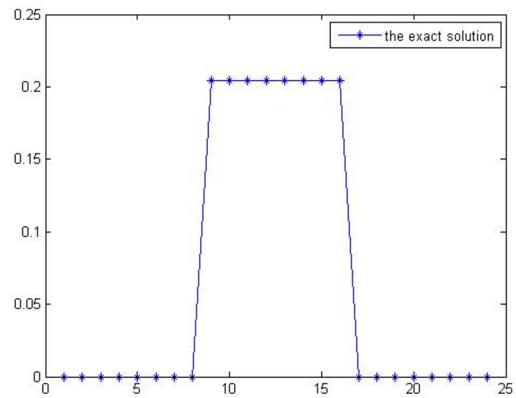
We also show all the x_k below in the two graphs:



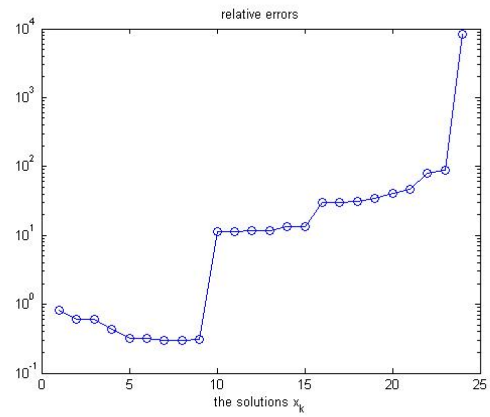


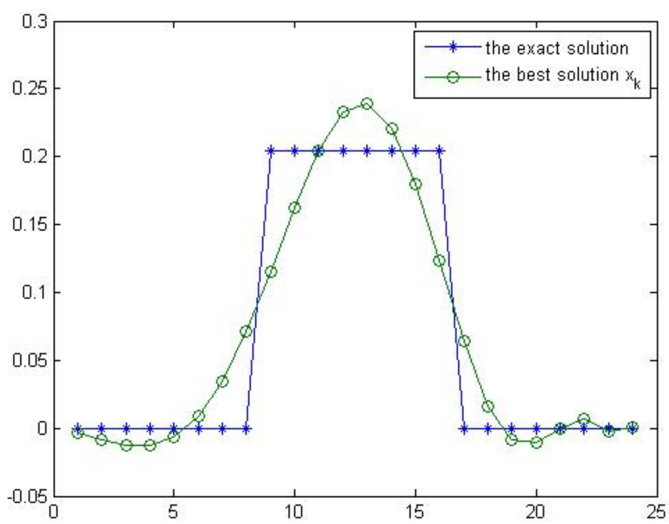
4.

We set $n = 24$ and use **wing** function, we plot the exact solution on the right, we can see that the exact solution is discontinuous. Then we will apply both TSVD and Tikhonov methods to see their limiatations.

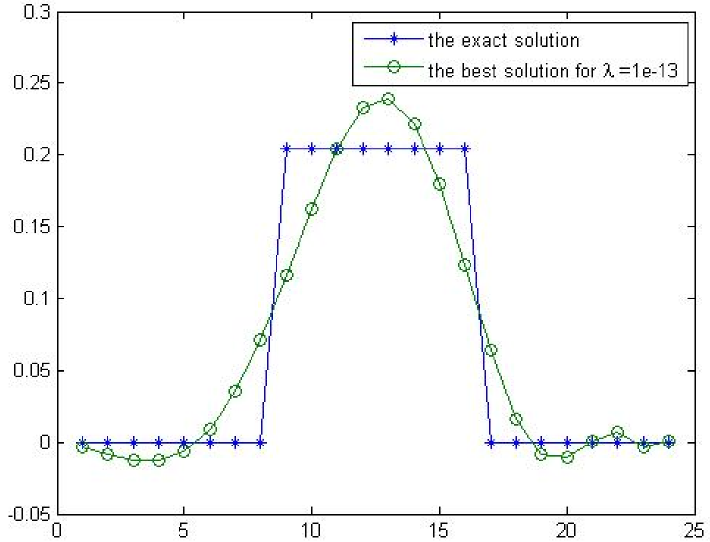


First, we compute it by TSVD, we compute the relative errors for each k from 1 to 24. The best k is 8, which makes the relative error have its minimum value 0.3023, but this is still very big, thus we can see the limitation of TSVD.





Then, we compute the problem using Tikhonov, we let λ be 10^{-15} to 10, then we get the best $\lambda = 10^{-13}$ to get the minimum relative error 0.3023, which is also very big, thus shows the limitation of Tikhonov.



References:

[1] Hansan's Notes Page 19-20

[2] Hansan's Notes Page 30

[3] Hansan's Notes Page 30-31

Matlab Commands:

```
%1
clc;clear
x(1)=1.5;
for i=1:100
    x(i+1)=(x(i)+1)^(1/3);
end
norm(x(100)-x(99))
x(100)


%2
clc;clear
format compact
[A,b] = ursell(12);
[U,S,V]=svd(A);
s=diag(S);
eta = picard(U,s,b);


clc;clear
clc;clear
%3.1
n=24;
```

```
[A b x]=shaw(n);
[U S V]=svd(A);
s=diag(S);
%3.2
semilogy(s,'*');
legend('singular value \sigma_i')
title('singular value of A')
min(find(s<10e-5));
for i=1:9
    subplot(3,3,i)
    plot(U(:,i));
    title(['i=',num2str(i)])
    ylabel('U(:,i)')
end
xlabel('left singular vectors')
figure
for i=1:9
    subplot(3,3,i)
    plot(U(:,i));
    title(['i=',num2str(i)])
    ylabel('V(:,i)')
end
%3.3
picard(U,s,b);
%3.4
r=rand(24,1);
a=norm(b)*(1e-10)/norm(r);
e=a*r;
bn=b+e;
picard(U,s,bn)
title('after adding noise to right
hand side b')
%3.5
xk=[];
rlterror=[];
for k=1:24
    xnew=tsvd(U,s,V,bn,k);
    xk=[xk,xnew];
    errnew=norm(x-xnew)/norm(x);
    rlterror=[rlterror,errnew];
end
plot(x,'-*');title('exact solution
x')
figure
semilogy(rlterror,'-o');
title('relative
errors');xlabel('the  solutions  x_k')
figure
for i=1:12
    subplot(4,3,i)
    plot(xk(:,i));
    title(['k=',num2str(i)])
end
figure
for i=13:24
    subplot(4,3,i-12)
    plot(xk(:,i));
    title(['k=',num2str(i)])
end
%3.6   3.6.4
aa=norm(b)*(1e-2)/norm(r);
ee=aa*r;
bnn=b+e;
picard(U,s,bnn)
title('after adding noise to right
hand side b')
%3.6   3.6.5
xkk=[];
rlterrork=[];
for k=1:24
    xnew=tsvd(U,s,V,bnn,k);
    xkk=[xkk,xnew];
    errnew=norm(x-xnew)/norm(x);
    rlterrork=[rlterrork,errnew];
end
semilogy(rlterrork,'-o');
title('relative
errors');xlabel('the  solutions  x_k')
figure
for i=1:12
    subplot(4,3,i)
    plot(xkk(:,i));
    title(['k=',num2str(i)])
end
figure
for i=13:24
    subplot(4,3,i-12)
    plot(xkk(:,i));
    title(['k=',num2str(i)])
end



%4
clc;clear
n=24;
[A,b,x]=wing(n);
[U S V]=svd(A);
s=diag(S);
plot(x,'-*')
legend('the exact solution')
% By TSVD
xk=[];
rlterror=[];
y=[1:n];
for k=1:n
    xnew=tsvd(U,s,V,b,k);
    xk=[xk,xnew];
    errnew=norm(x-xnew)/norm(x);
    rlterror=[rlterror,errnew];
end
figure
semilogy(rlterror,'-o');
```

```
title('relative
errors');xlabel('the  solutions  x_k')
figure
bestk=find(rlterror==min(rlterror));
rlterror(bestk);
plot(y,x,'-*',y,xk(:,bestk),'-o')
legend('the exact solution','the
best solution x_k')

% By Tikhonov
lambda=[1e-15,1e-13,1e-11,1e-9,1e-
7,1e-5,1e-3,0.1,1];
xl=[];
rlterror2=[];
for i=1:length(lambda)

[x_lambda,rho,eta]=tikhonov(U,s,V,b
,lambda(i));
    xl=[xl,x_lambda];
    errnew=norm(x-x_lambda)/norm(x);
    rlterror2=[rlterror2,errnew];
end
bestlmd=find(rlterror2==min(rlterro
r2))
rlterror2(bestlmd)
figure
plot(y,x,'-*',y,xl(:,bestlmd),'-o')
legend('the exact solution','the
best solution for \lambda =1e-13')
```