Contents

- CS375 HW12
- Problem 2c
- Problem 3a
- Problem 3b
- Problem 3c
- Problem 3d
- Problem 4c

CS375 HW12

Juan Alejandro Ormaza November 30, 2021


```
clear all;
clc;
close all;
```

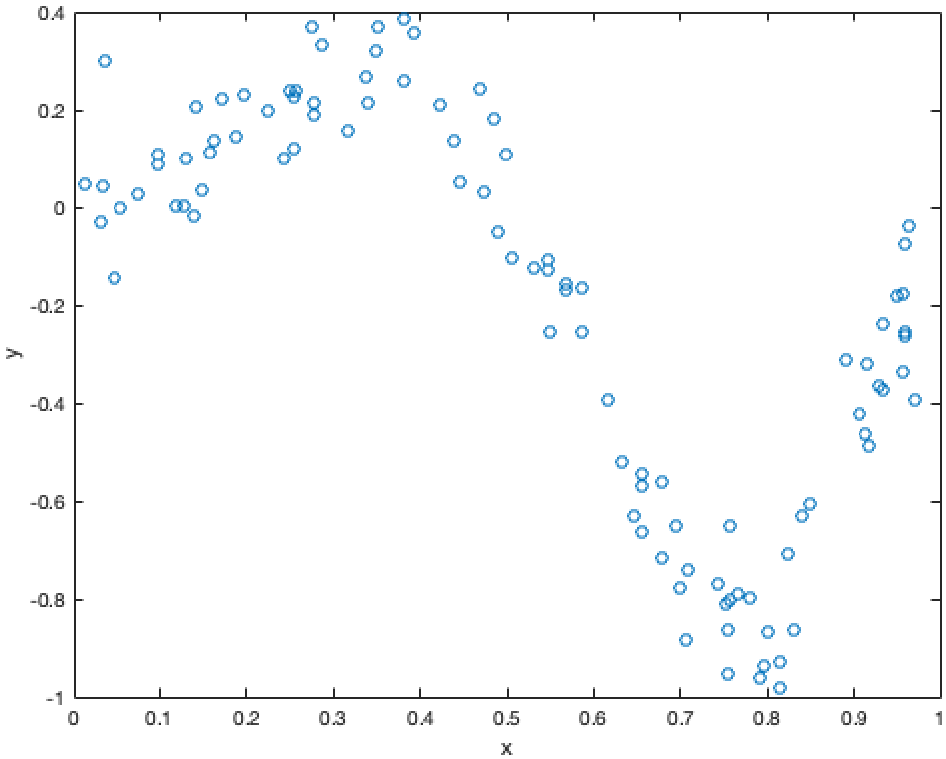
Problem 2c

```
A= [1 -1;2 3];
b=[1;1];

qr_solve(A,b)
```

```
ans =

    0.8000
   -0.2000
```

Problem 3a

```
N=100;

[x,y] = generate_ls_data(N);

figure();
plot(x,y,'o');
xlabel('x');
ylabel('y');
```

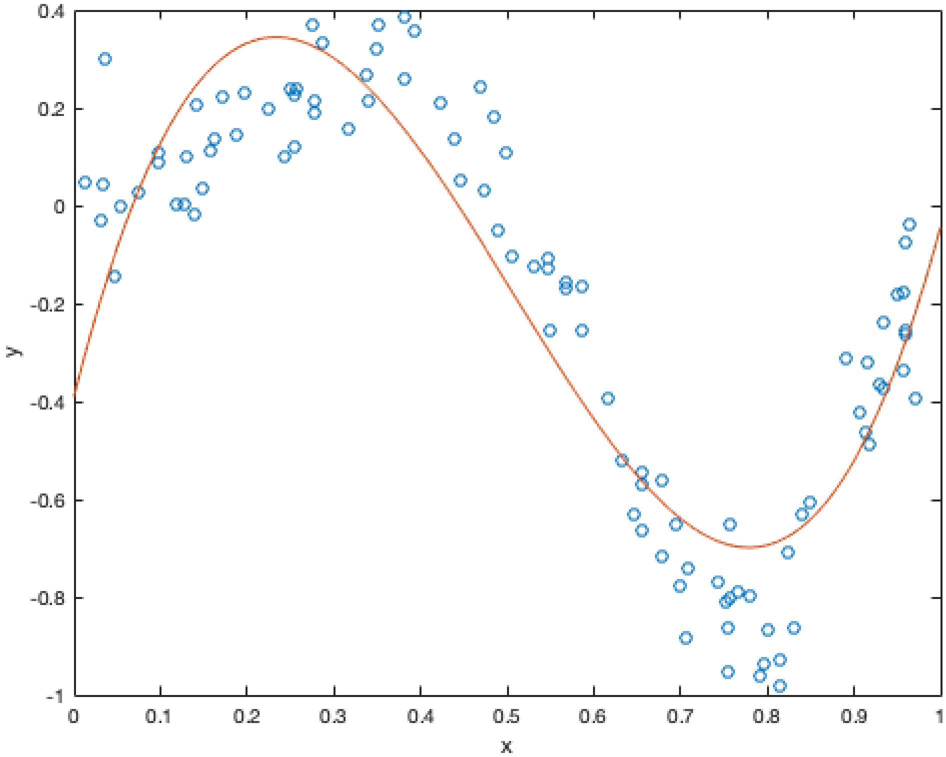


Problem 3b

```
% A'Ax=A'b

V=Vandermonde(x);

AtA=V'*V;
AtB=V'*y;
c=AtA\AtB;

coefficients = c;
coefficients=rot90(coefficients);
coefficients=rot90(coefficients);

xfine = linspace(0,1,1000);
yfine = polyval(coefficients,xfine);

figure();
plot(x,y,'o');
hold on
plot(xfine,yfine);
xlabel('x');
ylabel('y');

fprintf("the coefficients are: \n")
coefficients
```

```
the coefficients are:

coefficients =
```

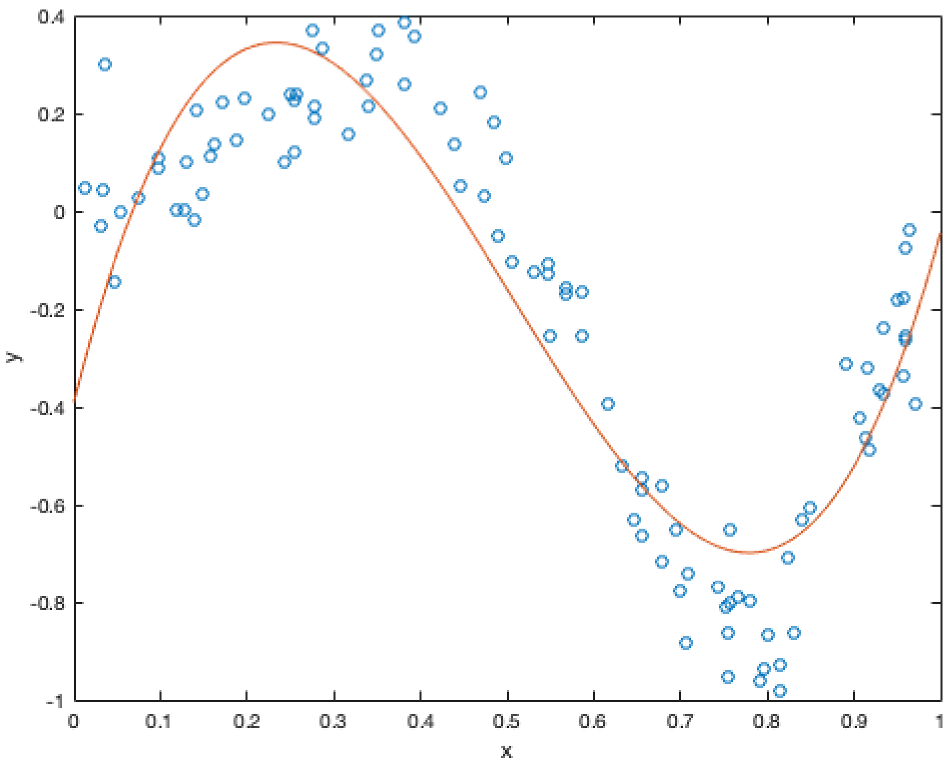
```
  12.8410
 -19.4916
   6.9961
  -0.3899
```



Problem 3c

```
b=y;
c2=qr_solve(V,b);

coefficients2 = c2(1:4);
coefficients2=rot90(coefficients2);
coefficients2=rot90(coefficients2);

xfine = linspace(0,1,1000);
yfine = polyval(coefficients2,xfine);

figure();
plot(x,y,'o');
hold on
plot(xfine,yfine);
xlabel('x');
ylabel('y');

fprintf("the coefficients are: \n")
coefficients2
```

```
the coefficients are:

coefficients2 =

   12.8410
  -19.4916
    6.9961
   -0.3899
```

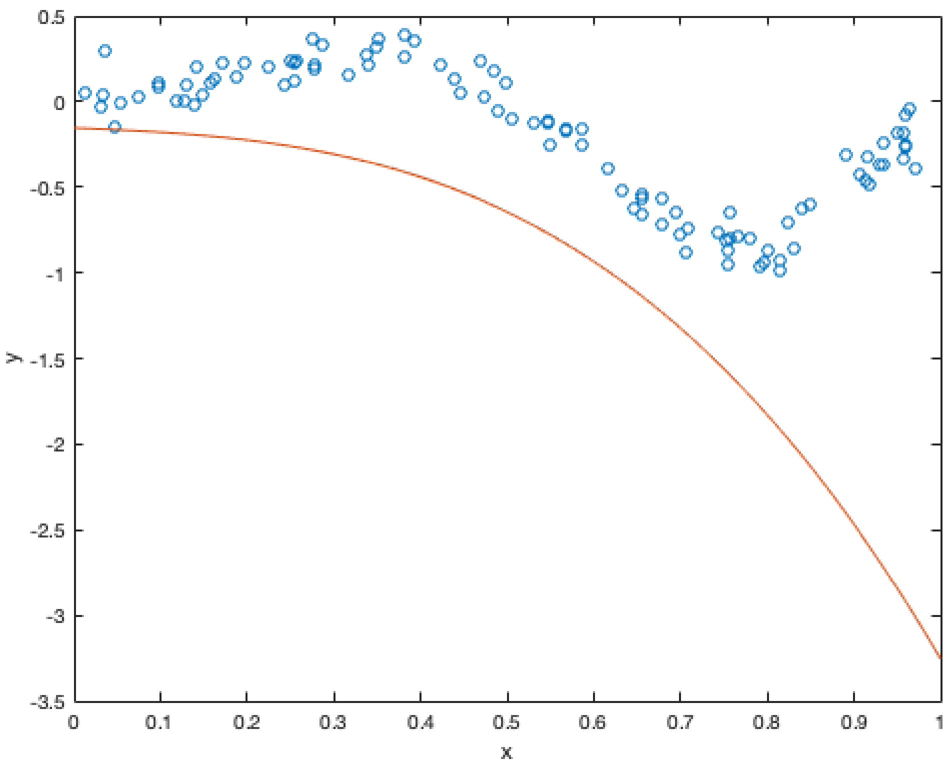


Problem 3d

```
A=V;

[U,S,V] = svd(A);
sigma=inv(S(1:4,1:4));
c3=zeros(4,1);

Ut=U';

for i=1:4
    c3(i)=sum((sigma(i,i)*Ut(i,i).*b)*V(i,i));
end


coefficients3 = c3(1:4);
coefficients3=rot90(coefficients3);
coefficients3=rot90(coefficients3);


xfine = linspace(0,1,1000);
```

```
yfine = polyval(coefficients3,xfine);

figure();
plot(x,y,'o');
hold on
plot(xfine,yfine);
xlabel('x');
ylabel('y');

fprintf("the coefficients are: \n")
coefficients3
```

```
the coefficients are:

coefficients3 =

   -2.6596
   -0.2422
   -0.2000
   -0.1527
```



Problem 4c

```
f=@(u,t) -u^2 - 2*sin(2*t) + (cos(2*t))^2;

a=0;
b=1;
u0=1;
```

```
n = [10, 20, 40, 80];
error=zeros(length(n),1);
valAt1=zeros(length(n),1);
h = (b-a)./n;

p=zeros(length(n),1);

for i=1:length(n)
    y=euler(n(i),a,b,u0,f);
    valAt1(i)=y(end);
    error(i)=abs(cos(2*1)-valAt1(i));
    if(i>1)
        p(i)=log(error(i)/error(i-1))/log(h(i)/h(i-1));
    end
end

fprintf("h\t\t approximation t=1\t error\t\t order of convergence p\n");
fprintf("%1.6f\t %1.6f\t\t %1.10f\t %2.3f\n",[h ;valAt1'; error'; p'])
```

```
h                 approximation t=1       error           order of convergence p
0.100000          -0.166287               0.2498599583    0.000
0.050000          -0.293181               0.1229661619    1.023
0.025000          -0.355195               0.0609516664    1.013
0.012500          -0.385809               0.0303381381    1.007
```

Published with MATLAB® R2021a

CS375 HW 12

Juan Alejandro Ormaza

Nov 30^{th} 2021

1.a) $|\det(BC)| = |\det(B)||\det(C)|$ (i)

The absolute value of determinant of a unitary (orthogonal in real arithmetic) matrix is 1. (ii)

The determinant of a diagonal matrix is the product of the diagonal entries (iii)

use these three facts to show that if $A \in \mathbb{R}^{n\times n}$ (square matrix) then

$$|\det(A)| = \prod_{i=1}^{n} \sigma_i$$

Where σ_i is the ith Singular value. In other words, the absolute value of the determinant of a square matrix is the product of its singular values.

$$A = USV^T$$

$$|\det(A)| = |\det(USV^T)|$$

Product of diagonal entries

$$|\det(A)| = \overset{1}{\cancel{|\det(U)|}}\,|\det(S)|\,\overset{1}{\cancel{|\det(V^T)|}}$$

Since V & U are orthogonal & $\underbrace{V^TV = I \quad U^TU = I}$

$U^{-1} = U^T$

$V^{-1} = V^T$

$$\boxed{|\det(A)| = \prod_{i=1}^{n} \sigma_i \qquad S = \sigma}$$

b)

$$A = \begin{bmatrix} 1 & -1 \\ -2 & 2 \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 1 & -2 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ -2 & 2 \end{bmatrix} = \begin{bmatrix} 5 & -5 \\ -5 & 5 \end{bmatrix}$$

$$\begin{vmatrix} 5-\lambda & -5 \\ -5 & 5-\lambda \end{vmatrix} = \lambda^2 - 10\lambda = \lambda(\lambda - 10)$$

$$\lambda_1 = 10$$

$$\lambda_2 = 0$$

$$S^2 = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix} \qquad S = \begin{bmatrix} \sqrt{10} & 0 \\ 0 & 0 \end{bmatrix}$$

$\lambda_1 = 10$

$$\begin{bmatrix} -5 & -5 \\ -5 & -5 \end{bmatrix} v_1 = 0 \rightarrow \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} v_1 = 0 \quad v_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

$\lambda_2 = 0$

$$\begin{bmatrix} 5 & -5 \\ -5 & 5 \end{bmatrix} v_2 = 0 \rightarrow \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} v_2 = 0 \quad v_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

$$V = \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

$$AA^T = \begin{bmatrix} 1 & -1 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} 1 & -2 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 2 & -4 \\ -4 & 8 \end{bmatrix}$$

$\lambda_1 = 10$

$$\begin{bmatrix} -8 & -4 \\ -4 & -2 \end{bmatrix} u_1 = 0 \rightarrow \begin{bmatrix} 2 & 1 \\ 0 & 0 \end{bmatrix} u_1 = 0$$

$$\begin{bmatrix} 1 & 1/2 \\ 0 & 0 \end{bmatrix} u_1 = 0$$

$$u_1 = \begin{bmatrix} 1/2 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{5} \\ -2/\sqrt{5} \end{bmatrix}$$

$\lambda_2 = 0$

$$\begin{bmatrix} 2 & -4 \\ -4 & 8 \end{bmatrix} u_2 = 0 \Rightarrow \begin{bmatrix} 1 & -2 \\ 0 & 0 \end{bmatrix} u_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$u_2 = \begin{bmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}$$

$$U = \begin{bmatrix} 1/\sqrt{5} & 2/\sqrt{5} \\ -2/\sqrt{5} & 1/\sqrt{5} \end{bmatrix}$$

$$\boxed{A = \begin{bmatrix} 1/\sqrt{5} & 2/\sqrt{5} \\ -2/\sqrt{5} & 1/\sqrt{5} \end{bmatrix} \begin{bmatrix} \sqrt{10} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}}$$

2. a) Assume you are given the QR factorization of an nxn invertible matrix, i.e., $A = QR$, where Q is unitary (orthogonal in real arithmetic) and R is upper triangle. Explain each step for using Q and R to solve $Ax=b$ for x

to solve $Ax=b$ with $A=QR$

we want to first solve $Rx=C$ using backward substitution. where $C=Q^Tb$

this can be seen when we plug $A=QR$ into $Ax=b$

$$QRx=b$$

$$\underbrace{Q^TQ}_{I}\underbrace{Rx}_{C}=Q^Tb$$

$$IC=Q^Tb \quad C=Q^Tb$$

4. a) Suppose that an ordinary differential equation is solved numerically on an interval $[a, b]$ and that the local truncation error is ch^p. Show that if all truncation errors have the same sign (the worst possible case), then the total truncation error is $(b-a)ch^{p-1}$, where $h = \frac{b-a}{n}$

$$\text{Total error} = \sum_{i=1}^{n} ch^p = ch^p + ch^p + ch^p + \dots + ch^p = nch^p$$

$$n = \frac{b-a}{h}$$

$$\boxed{\frac{b-a}{h} ch^p = (b-a)ch^{p-1}}$$

b) By hand) consider the following ODE:

$$\begin{cases} u' = -u^2 - 2\sin(2t) + \cos^2(2t) & t \in [0,1] \quad (1) \\ u(0) = 1 \end{cases}$$

verify that $u(t) = \cos(2t)$ satisfies both the ODE and the initial conditions

ODE:

Plug in (1)

$$u(t) = \cos(2t)$$

$$u'(t) = -2\sin(2t)$$

$$-2 \cdot \sin(2t) = \cancel{-\cos^2(2t)} - 2\sin(2t) \cancel{+\cos^2(2t)}$$

$$-2\sin(2t) = -2\sin(2t) \checkmark$$

$\cos(2t)$ is a solution

initial conditions:

$U(0) = 1$

$u(t=0) = \cos(2 \cdot 0) = 1$

$u'(t=0) = -2\sin(2 \cdot 0) = 0$

$u' = -u^2 - 2\sin(2t) + \cos^2(2t)$

$0 = \cancel{-1} - 0 + \cancel{1}$

$0 = 0$

initial conditions are satisfied ∴ $u(t) = \cos(2t)$ Satisfies both the ODE and the initial condition.

```
% CS375 qr solve
% Juan Alejandro Ormaza
% November 30, 2021

function x = qr_solve(A,b)

[Q,R] = qr(A);

C=Q'*b;

x=R\C;

end
```

```
Not enough input arguments.

Error in qr_solve (line 7)
[Q,R] = qr(A);
```

Published with MATLAB® R2021a

```
function [x,y] = generate_ls_data(N)

% Set the random seed to get reproducible results
rng('default');

% Generate x-data uniformly distributed over [0,1]
x = rand(N,1);

% Generate y = x*sin(2\pi x) + noise
y = x.*sin(2*pi*x) + 0.1*randn(N,1);
```

```
Not enough input arguments.

Error in generate_ls_data (line 7)
x = rand(N,1);
```

Published with MATLAB® R2021a

```
% CS375 Vandermonde
% Juan Alejandro Ormaza
% November 30, 2021

function [V] = Vandermonde(x)

n=length(x)-1;
V=zeros(n+1,4);

for i=1:n+1
    for j=1:4
        if(j==1)
            V(i,j)=1;
        else
            V(i,j)=x(i)^(j-1);
        end
    end
end

end
```

```
Not enough input arguments.

Error in Vandermonde (line 7)
n=length(x)-1;
```

Published with MATLAB® R2021a

```
% CS375 euler
% Juan Alejandro Ormaza
% November 30, 2021

function [u] = euler(n,a,b,u0,f)

h=(b-a)/n;
u=zeros(n,1);
t=a:h:b;
u(1)=u0;
for i=1:n-1
    u(i+1)=u(i)+ h*f(u(i),t(i));
end
```

```
Not enough input arguments.

Error in euler (line 7)
h=(b-a)/n;
```

Published with MATLAB® R2021a