Name: PARMESH YADAV

Roll No: 2020319

SCIENTIFIC COMPUTING (MTH373)

HOMEWORK – 2

Problem – 1

Problem-1

(a) $x_1^2 + \cancel{2x_2^2} + 3x_3^2 + (x_1 - x_2 + x_3 - 1)^2 + (-x_1 - 4x_2 + 2)^2 = f(x)$

$$\nabla f = \begin{bmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_2 \\ \partial f/\partial x_3 \end{bmatrix} = 0$$

$\therefore \dfrac{\partial f}{\partial x_1} = 6x_1 + 6x_2 + 2x_3 - 6$

$\therefore \dfrac{\partial f}{\partial x_2} = 6x_1 + 18x_2 - 2x_3 - 14$

$\therefore \dfrac{\partial f}{\partial x_3} = 2x_1 - 2x_2 + 8x_3 - 2$

$\therefore \nabla f = 0$

$$\Rightarrow \underbrace{\begin{bmatrix} 6 & 6 & 2 \\ 6 & 18 & -2 \\ 2 & -2 & 8 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{x} \cong \underbrace{\begin{bmatrix} 6 \\ 14 \\ 2 \end{bmatrix}}_{b}$$

(b) $f(x) = (-6x_2 + 4)^2 + (-4x_1 + 3x_2 - 1)^2 + (x_1 + 8x_2 - 3)^2$

$$\nabla f = \begin{bmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_2 \end{bmatrix} = 0$$

$\therefore \dfrac{\partial f}{\partial x_1} = 34x_1 - 8x_2 + 2$

$\therefore \dfrac{\partial f}{\partial x_2} = 218x_2 - 8x_1 - 102$

$\therefore \nabla f = 0$

$$\Rightarrow \underbrace{\begin{bmatrix} 34 & -8 \\ -8 & 218 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x} \cong \underbrace{\begin{bmatrix} -2 \\ 102 \end{bmatrix}}_{B}$$

(c) $f(x) = 2(-6x_2 + 4)^2 + 3(-4x_1 + 3x_2 - 1)^2 + 4(x_1 + 8x_2 - 3)^2$

$$\nabla f = \begin{bmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_2 \end{bmatrix} = 0$$

$\therefore \dfrac{\partial f}{\partial x_1} = 104x_1 - 8x_2$

$\therefore \dfrac{\partial f}{\partial x_2} = -6x_1 + 710x_2 - 306$

$\therefore \nabla f = 0$

$$\Rightarrow \underbrace{\begin{bmatrix} 104 & -8 \\ -6 & 710 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x} \cong \underbrace{\begin{bmatrix} 0 \\ 306 \end{bmatrix}}_{B}$$

Problem – 2

Problem-2

$$\begin{bmatrix}1 & 1\\0 & 1\\0 & 0\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} \approx \begin{bmatrix}2\\1\\1\end{bmatrix}$$

$$A \quad x \approx B$$

Using necessary condition for existence i.e. normal equations,

$$A^TAx = A^Tb$$

$$\therefore A^TA = \begin{bmatrix}1 & 0 & 0\\1 & 1 & 0\end{bmatrix}\begin{bmatrix}1 & 1\\0 & 1\\0 & 0\end{bmatrix} = \begin{bmatrix}1 & 1\\1 & 2\end{bmatrix}$$

$$\therefore A^Tb = \begin{bmatrix}1 & 0 & 0\\1 & 1 & 0\end{bmatrix}\begin{bmatrix}2\\1\\1\end{bmatrix} = \begin{bmatrix}2\\3\end{bmatrix}$$

$$\therefore \begin{bmatrix}1 & 1\\1 & 2\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}2\\3\end{bmatrix}$$

$$\Rightarrow x_1 + x_2 = 2 \quad \text{(1)}$$
$$x_1 + 2x_2 = 3 \quad \text{(2)}$$

We get, $x_1 = 1$, $x_2 = 1$ $\Rightarrow x = \begin{bmatrix}1\\1\end{bmatrix}$

To verify this as the minimum solⁿ, we need to use the sufficiency condition

$$\nabla^2 f = 2A^TA > 0$$

$$2A^TA = \begin{bmatrix}2 & 2\\2 & 4\end{bmatrix} \Rightarrow \|2A^TA\|_2 > 0$$

Now, to compute norm of min. residual vector.

$$\|b - Ax\|_2$$

$$= \left\| \begin{bmatrix}2\\1\\1\end{bmatrix} - \begin{bmatrix}1 & 1\\0 & 1\\0 & 0\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix} \right\|_2$$

$$= \left\| \begin{bmatrix}2\\1\\1\end{bmatrix} - \begin{bmatrix}2\\1\\0\end{bmatrix} \right\|_2$$

$$= \left\| \begin{bmatrix}0\\0\\1\end{bmatrix} \right\|_2$$

$$= 1$$

Problem – 3

Problem-3 $A \in R^{m \times n}$, $m \geq n$, linearly independent columns

(a) $\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix}$: order $= (m+n) \times (m+n)$

Determinant : $(I \times 0) - (A^T \times A^T)$

: $\cancel{AA^T}$ $- A^T A$

Now, for any $x \neq 0 \in R^n$,

$Ax \neq 0$

$\therefore$ Let $y = A^T A x$

$\Rightarrow x^T A^T A x = x^T y = \|Ax\|_2^2 \neq 0$

$\Rightarrow y \neq 0$

$\Rightarrow A^T A x \neq 0$

$\Rightarrow A^T A$ is non-singular

$\therefore$ The given matrix is also non-singular.

(b) $\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$

$\Rightarrow \begin{bmatrix} \hat{x} + A\hat{y} \\ A^T \hat{x} \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$

This gives us, $\hat{x} + A\hat{y} = b$ — (1)

$A^T . \hat{x} = 0$ — (2)

$\Rightarrow \hat{x}^T A = 0$ [Transpose]

$\therefore$ We know, $r^T A = 0$ [Residual from LS]

$\Rightarrow \hat{x} = r$

$\boxed{\hat{x} = b - Ax}$ — (3)

Further, put (3) in (1), $\boxed{\hat{y} = x}$

Problem – 5

a)

Problem - 5

(a) $y = f(t) = \dfrac{e^{\alpha t+\beta}}{1+e^{\alpha t+\beta}}$

$\Rightarrow \quad y[1+e^{\alpha t+\beta}] = e^{\alpha t+\beta}$

$y + ye^{\alpha t+\beta} = e^{\alpha t+\beta}$

$y = e^{\alpha t+\beta}[1-y]$

$e^{\alpha t+\beta} = \dfrac{y}{1-y}$

Taking log both sides,

$\log[e^{\alpha t+\beta}] = \log\left[\dfrac{y}{1-y}\right]$

$-[\alpha t+\beta] = \log\left[\dfrac{1-y}{y}\right]$

$-[\alpha t+\beta] = \log\left[\dfrac{1}{y} - 1\right]$

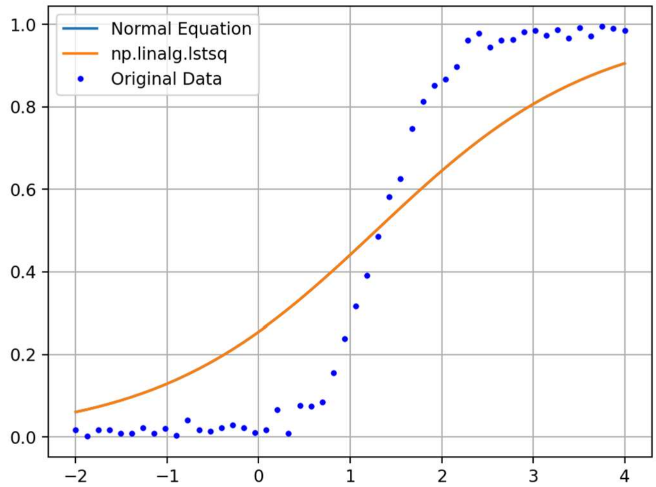
Therefore, for m points,

$$x = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}_{2\times 1}, \quad B = \begin{bmatrix} \log[(1/y_1) - 1] \\ \vdots \\ \log[(1/y_m) - 1] \end{bmatrix}_{m\times 1}$$

$$A = \begin{bmatrix} -t_1 & -1 \\ -t_2 & -1 \\ \vdots & \vdots \\ -t_m & -1 \end{bmatrix}_{m\times 2}$$

b)

```
[Running] python -u "d:\College\Sem 5\Scientific C
Solved Using Normal Equation
Alpha = 0.833341359611429
Beta = -1.071111397147609
2-Norm Residual = 3.0315184435401137

Solved Using np.linalg.lstsq
Alpha = 0.8333413596114293
Beta = -1.0711113971476087
2-Norm Residual = 3.0315184435401132
```



As the difference between the 2 solutions, one by normal equations and the other by np.linalg.lstsq is very low, the plots overlap, hence, only one is visible.

Problem – 6

Problem-6

(a) $A = \begin{bmatrix} 1 & 1 \\ 10^{-k} & 0 \\ 0 & 10^{-k} \end{bmatrix}$, $b = \begin{bmatrix} -10^{-k} \\ 1+10^{-k} \\ 1-10^{-k} \end{bmatrix}$

$$\Rightarrow A^T A = \begin{bmatrix} 1 & 10^{-k} & 0 \\ 1 & 0 & 10^{-k} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 10^{-k} & 0 \\ 0 & 10^{-k} \end{bmatrix}$$

$$A^T A = \begin{bmatrix} 1+10^{-2k} & 1 \\ 1 & 1+10^{-2k} \end{bmatrix}$$

$$\Rightarrow A^T b = \begin{bmatrix} 1 & 10^{-k} & 0 \\ 1 & 0 & 10^{-k} \end{bmatrix} \begin{bmatrix} -10^{-k} \\ 1+10^{-k} \\ 1-10^{-k} \end{bmatrix} = \begin{bmatrix} 10^{-2k} \\ -10^{-2k} \end{bmatrix}$$

By using, $A^T A x = A^T b$,

$$\begin{bmatrix} 1+10^{-2k} & 1 \\ 1 & 1+10^{-2k} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 10^{-2k} \\ -10^{-2k} \end{bmatrix}$$

Solving this 2×2 linear system, we get

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Using QR Decomposition (6_B)

```
k = 6
x = [[ 1.]
 [-1.]]

k = 7
x = [[ 1.]
 [-1.]]

k = 8
x = [[ 1.00000001]
 [-1.00000001]]

k = 9
x = [[ 1.00000005]
 [-1.00000005]]

k = 10
x = [[ 1.00000052]
 [-1.00000052]]

k = 11
x = [[ 1.00002329]
 [-1.00002329]]

k = 12
x = [[ 0.99991338]
 [-0.99991338]]

k = 13
x = [[ 0.99936385]
 [-0.99936385]]

k = 14
x = [[ 1.01270964]
 [-1.01270964]]

k = 15
x = [[ 0.86355085]
 [-0.86355085]]
```

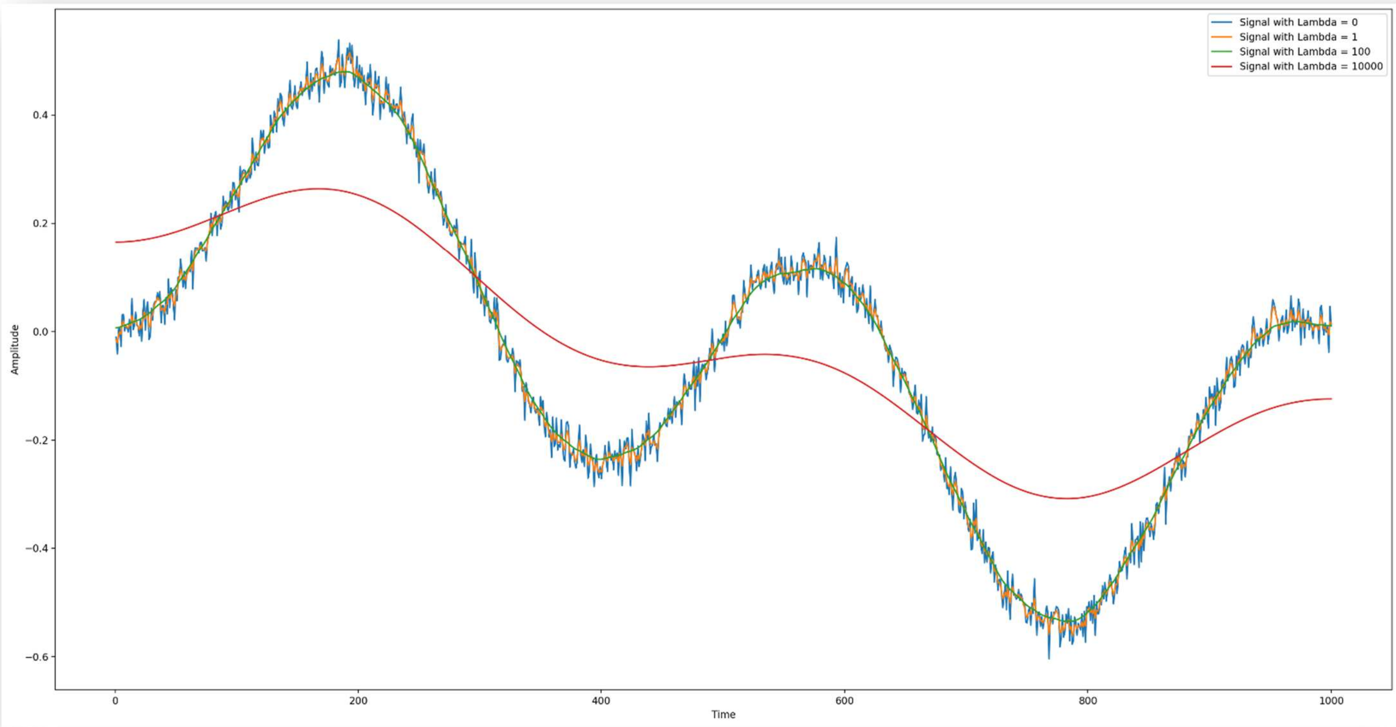
Using Normal Equations(6_c)

```
k = 6
x = [[ 0.99991111]
 [-0.99991111]]

k = 7
x = [[ 1.00079992]
 [-1.00079992]]

k = 8
Traceback (most recent call last):
  File "d:\College\Sem 5\Scientific Computing\HomeWork\HomeWork 2\problem_6c.py
    start()
  File "d:\College\Sem 5\Scientific Computing\HomeWork\HomeWork 2\problem_6c.py
    x = np.linalg.solve(A_t_A, A_t_b)
  File "<__array_function__ internals>", line 180, in solve
  File "C:\Users\PARMESH YADAV\AppData\Local\Programs\Python\Python310\lib\site
    r = gufunc(a, b, signature=signature, extobj=extobj)
  File "C:\Users\PARMESH YADAV\AppData\Local\Programs\Python\Python310\lib\site
    raise LinAlgError("Singular matrix")
numpy.linalg.LinAlgError: Singular matrix
```

As we can see in both the outputs, as the value of K increases, the perturbation in the final answers also increases. In the second case i.e., solving using Normal Equation, for K >= 8, the matrix A^TA becomes singular.

Problem – 7

Output



As we can see in the output, the blue line represents the original signal and the orange, green and red represent the 3 reconstructed signals for lambda = 1,100,10000.

We can observe that for lambda = 1 [Orange], the reconstruction is not very useful, it's still having noise in it. The most accurate reconstruction is for lambda = 100 [Green] and as lambda increases to lambda = 10000 [Red], the output doesn't have noise but is perturbed very much from the original signal.