ECE 367 PSet 4

5.1)

$A = \frac{1}{\sqrt{10}}\begin{bmatrix}5 & 0\\ 3 & 4\end{bmatrix}$

a) $\sigma_i(A) = \sqrt{\lambda_i(AA^T) \neq 0}$

$AA^T = \frac{1}{\sqrt{10}}\begin{bmatrix}5 & 0\\ 3 & 4\end{bmatrix}\frac{1}{\sqrt{10}}\begin{bmatrix}5 & 3\\ 0 & 4\end{bmatrix} = \frac{1}{10}\begin{bmatrix}25 & 15\\ 15 & 25\end{bmatrix} = \begin{bmatrix}5/2 & 3/2\\ 3/2 & 5/2\end{bmatrix}$

$\det(AA^T - \lambda I) = 0 \Rightarrow (\lambda - \frac{5}{2})^2 - (\frac{3}{2})^2 = 0 \Rightarrow (\lambda - \frac{5}{2} + \frac{3}{2})(\lambda - \frac{5}{2} - \frac{3}{2}) = 0 \Rightarrow (\lambda - 1)(\lambda - 4) = 0 \Rightarrow \lambda_1 = 4, \lambda_2 = 1$

This gives $\sigma_1 = 2, \sigma_2 = 1$

$\Sigma = \begin{bmatrix}\sigma_1 & 0\\ 0 & \sigma_2\end{bmatrix} \Rightarrow \boxed{\Sigma_1 = \begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}}$

Right singular vectors = eigenvectors of A^TA

$A^TA = \frac{1}{10}\begin{bmatrix}5 & 3\\ 0 & 4\end{bmatrix}\begin{bmatrix}5 & 0\\ 3 & 4\end{bmatrix} = \frac{1}{10}\begin{bmatrix}34 & 12\\ 12 & 16\end{bmatrix}$

Since $\lambda_i(A^TA) \neq 0 = \lambda_i(AA^T) \neq 0$; $\lambda(A^TA) = 4, 1$

$\frac{1}{10}\begin{bmatrix}34 & 12\\ 12 & 16\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}4x\\ 4y\end{bmatrix}$

$\left.\begin{array}{l}3.4x + 1.2y = 4x\\ 1.2x + 1.6y = 4y\end{array}\right\} x = 2y,\ v_1 = \begin{bmatrix}2\\ 1\end{bmatrix}/\sqrt{5}$

$\frac{1}{10}\begin{bmatrix}34 & 12\\ 12 & 16\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}x\\ y\end{bmatrix}$

$\left.\begin{array}{l}3.4x + 1.2y = x\\ 1.2x + 1.6y = y\end{array}\right\} y = -2x \quad v_2 = \begin{bmatrix}1\\ -2\end{bmatrix}/\sqrt{5}$

$V = [v_1\ v_2] = \begin{bmatrix}2/\sqrt{5} & 1/\sqrt{5}\\ 1/\sqrt{5} & -2/\sqrt{5}\end{bmatrix} \Rightarrow \boxed{V = \begin{bmatrix}2/\sqrt{5} & 1/\sqrt{5}\\ 1/\sqrt{5} & -2/\sqrt{5}\end{bmatrix}}$

Matrix of normalized left-singular vectors

$AV = U\Sigma$

$AV = \frac{1}{\sqrt{10}}\begin{bmatrix}5 & 0\\ 3 & 4\end{bmatrix}\begin{bmatrix}2/\sqrt{5} & 1/\sqrt{5}\\ 1/\sqrt{5} & -2/\sqrt{5}\end{bmatrix} = \frac{1}{5\sqrt{2}}\begin{bmatrix}5 & 0\\ 3 & 4\end{bmatrix}\begin{bmatrix}2 & 1\\ 1 & -2\end{bmatrix} = \frac{1}{5\sqrt{2}}\begin{bmatrix}10 & 5\\ 10 & -5\end{bmatrix} = \begin{bmatrix}2/\sqrt{2} & 1/\sqrt{2}\\ 2/\sqrt{2} & -1/\sqrt{2}\end{bmatrix}\underbrace{\begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}}_{\Sigma} = U\Sigma$

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

b) $A = U\tilde{\Sigma}V^T = U\Sigma_1 V^T$

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

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

c) $Ax = \sigma_1 u^{(1)}\bar{x}_1 + \sigma_2 u^{(2)}\bar{x}_2 = 2u^{(1)}\bar{x}_1 + u^{(2)}\bar{x}_2$

$\|Ax\|_2^2 = \langle Ax, Ax\rangle$

$= (Ax)^T Ax$

$= (2\bar{x}_1 u^{(1)T} + \bar{x}_2 u^{(2)T})(2\bar{x}_1 u^{(1)} + \bar{x}_2 u^{(2)})$

$= 4\bar{x}_1^2 u^{(1)T}u^{(1)} + 2\bar{x}_1 \cancel{u^{(1)T}} u^{(2)} + 2\bar{x}_1\bar{x}_2 \cancel{u^{(2)T}} u^{(1)} + \bar{x}_2^2 u^{(2)T}u^{(2)}$

$= 4\bar{x}_1^2 + \bar{x}_2^2$

$\|x\|^2 = 1 \longrightarrow \|\bar{x}\| = 1 \longrightarrow \bar{x}_1^2 + \bar{x}_2^2 = 1$

$\therefore \|Ax\|_2^2 = 1 + 3\bar{x}_1^2, \quad 0 \le x_1 \le 1$

c) Max Amplification: $\bar{x}_1 = 1$

Input vector: $\bar{x} = \begin{bmatrix}1\\ 0\end{bmatrix} \to x = V\bar{x} = v^{(1)} \to \boxed{x = \begin{bmatrix}2/\sqrt{5}\\ 1/\sqrt{5}\end{bmatrix}}$

Amplification: $\boxed{\sigma_{max} = 2}$

Output direction: $\hat{Ax} = 2u^{(1)}(1) + u^{(2)}(0) = 2\hat{u}^{(1)} = \boxed{u^{(1)}}$

d) Min amplification: $\bar{x}_1 = 0$

Input vector: $\bar{x} = \begin{bmatrix}0\\ 1\end{bmatrix} \to x = V\bar{x} = v^{(2)} \to \boxed{x = \begin{bmatrix}1/\sqrt{5}\\ -2/\sqrt{5}\end{bmatrix}}$

Amplification: $\boxed{\sigma_{min} = 1}$

Output direction: $\hat{Ax} = 2u^{(1)}(0) + u^{(2)}(0) = \boxed{u^{(2)}}$

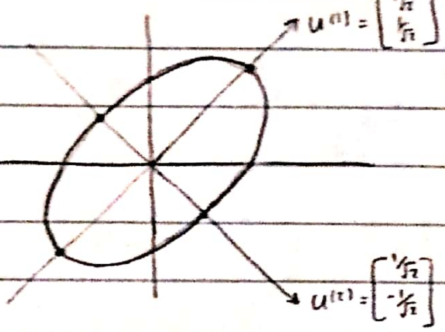
e) $\{Ax \in \mathbb{R}^2 : \|x\|_2 \le 1, x \in \mathbb{R}^2\}$

$Ax = 2\bar{x}_1 u^{(1)} + \bar{x}_2 u^{(2)}$

Since $u^{(1)}, u^{(2)}$ linearly independent, Ax can be any vector in $\mathbb{R}^2$, if no conditions on $\bar{x}_1$ and $\bar{x}_2$.

So, $Ax = \alpha_1 u^{(1)} + \alpha_2 u^{(2)}$

The condition $\|\bar{x}\| = \|x\| = 1$ gives $\left(\frac{\alpha_1}{2}\right)^2 + \alpha_2^2 = 1$, so $\{Ax \in \mathbb{R}^2 : \|x\|_2 \le 1, x \in \mathbb{R}^2\}$ defines an ellipse in $\mathbb{R}^2$, with basis vectors $u^{(1)}$ and $u^{(2)}$

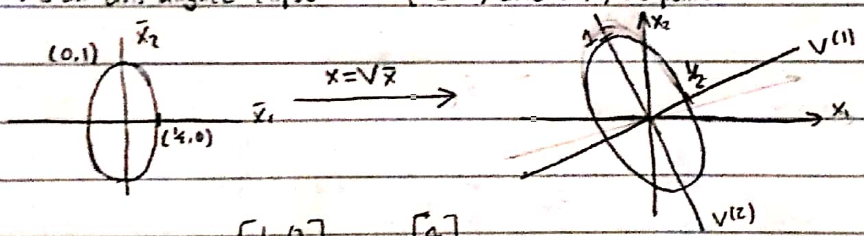


f) $\{x \in \mathbb{R}^2 : \|Ax\|_2 \le 1\}$

$\|Ax\|_2^2 = 4\bar{x}_1^2 + \bar{x}_2^2$

$4\bar{x}_1^2 + \bar{x}_2^2 \le 1$

This is an axis-aligned elipse with $(\frac{1}{2}, 0)$ and $(0,1)$ as points



5.2) $\vec{r} = A\vec{\alpha} - \vec{y} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} - \begin{bmatrix} 4 \\ 1 \\ 4 \\ 2 \end{bmatrix}$

Want to minimize $\|\vec{r}\|$ → equivalent to the following least-squares problem:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 4 \\ 2 \end{bmatrix}$$

Normal eqn: $A^T A \alpha^* = A^T y$

$$\alpha^* = (A^T A)^{-1} A^T y$$

$$\boxed{\alpha^* = \begin{bmatrix} 3.2 \\ -0.3 \end{bmatrix}} \text{ (Matlab)}$$

Plot in Matlab

Residuals: $r_1 = 0.8$, $r_2 = -1.9$, $r_3 = 1.4$, $r_4 = -0.3$ (Matlab)

Residual norm squared: $\|\vec{r}\|_2^2 = 6.3$ (Matlab)

b) $a=b=3$

$$W=\begin{bmatrix}1&0&0&0\\0&3&0&0\\0&0&3&0\\0&0&0&1\end{bmatrix}$$

$$\min_x \|Wr\|^2 = \min_x \|W(Ax-y)\|^2 = \min_x \|\tilde{A}x-\tilde{y}\|^2$$

where $\tilde{A}=WA$ and $\tilde{y}=Wy$

This is the standard LS problem in terms of $(\tilde{A},\tilde{y})$, meaning the solution x^* must satisfy normal equation:

$$\tilde{A}^T\tilde{A}x^*=\tilde{A}^T\tilde{y}$$

$\text{rank}(\tilde{A}^T\tilde{A})=\text{rank}(\tilde{A})=\text{rank}(WA)=\min(\text{rank}(W),\text{rank}(A))=\min(4,2)=2 \Rightarrow \tilde{A}^T\tilde{A}$ full rank since $\tilde{A}^T\tilde{A}\in\mathbb{R}^{2\times 2}$

$$x^*=(\tilde{A}^T\tilde{A})^{-1}\tilde{A}^T\tilde{y}$$
$$=(A^TW^TWA)^{-1}A^TW^TWy$$

$$\boxed{x^*=\begin{bmatrix}0.8000\\1.1667\end{bmatrix}} \quad (\text{Matlab})$$

Plot: See Matlab code

Residuals: $r_1=3.2$, $r_2=-0.9667$, $r_3=0.8667$, $r_4=-2.3$

Residual norm squared: $\|\vec{r}\|^2=17.2156$

The line in part b is closer to the points $(1,1)$ and $(2,4)$ than the line in part a, but further away from points $(0,4)$ and $(3,2)$. Comparing residuals, we have:

$$r_a=\begin{bmatrix}0.8\\-1.9\\1.4\\0.3\end{bmatrix} \qquad r_b=\begin{bmatrix}3.2\\-0.9667\\0.8667\\-2.3\end{bmatrix}$$

Notice that since $W=\text{diag}(1,3,3,1)$, it is important in part b to minimize the error in the 2^{nd} and 3^{rd} components. So, in part b, the line gives much higher residuals for r_1 and r_4 and lower residuals for r_2 and r_3, since these are 3 times more important

c) $a=4, b=2$

$$W=\begin{bmatrix}1&0&0&0\\0&3&1&0\\0&1&3&0\\0&0&0&1\end{bmatrix}$$

$$\min_{x\in\mathbb{R}^2}\|Wr\|_2^2=\min_{x\in\mathbb{R}^2}\|W(Ax-y)\|_2^2 \Rightarrow x^*=(A^TW^TWA)^{-1}A^TW^TWy$$

Substituting: $\boxed{x^*=\begin{bmatrix}1.8371\\0.4615\end{bmatrix}}$

Plot: See Matlab

Residuals: $r_1=2.1629$, $r_2=-1.2986$, $r_3=1.2398$, $r_4=-1.2217$

Residual Norm Squared: $\|\vec{r}\|^2=9.3943$

The difference between $\vec{r}_b$ and $\vec{r}_c$ is that residuals r_1 and r_4 have decreased, while r_2 and r_3 have increased.

$$\vec{r}_b=\begin{bmatrix}3.2\\-0.9667\\0.8667\\-2.3\end{bmatrix} \qquad \vec{r}_c=\begin{bmatrix}2.1629\\-1.2986\\1.2398\\-1.2217\end{bmatrix}$$

This suggest r_2 and r_3 are less important than they were previously. To show this:

$$Wr = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} r_1 \\ 3r_2+r_3 \\ 3r_3+r_2 \\ r_4 \end{bmatrix}$$

To minimize $\|Wr\|_2^2$, we want r_2 and r_3 to be opposite in sign, so $|3r_2+r_3|<|3r_2|$ and $|3r_3+r_2|<|3r_3|$.

We can write:

$$Wr = \begin{bmatrix} r_1 \\ 2r_2 \\ 2r_3 \\ r_4 \end{bmatrix} + \begin{bmatrix} 0 \\ r_2+r_3 \\ r_2+r_3 \\ 0 \end{bmatrix}$$

$$\|Wr\|^2 = r_1^2 + (2r_2+(r_2+r_3))^2 + (2r_3+(r_2+r_3))^2 + r_4^2$$

$$= r_1^2 + 4r_2^2 + 4r_2(r_2+r_3) + (r_2+r_3)^2 + 4r_3^2 + 4r_3(r_2+r_3) + (r_2+r_3)^2 + r_4^2$$

$$= r_1^2 + 4r_2^2 + 4(r_2+r_3)^2 + (r_2+r_3)^2 + 4r_3^2 + (r_2+r_3)^2 + r_4^2$$

$$= r_1^2 + 4r_2^2 + 6(r_2+r_3)^2 + 4r_3^2 + r_4^2$$

We want each term to be ≈ 0. So we want $r_i \approx 0 \ \forall i \in \{1,2,3,4\}$ and $r_2+r_3 \approx 0$. So $r_2 \approx -r_3$. This goes

$$Wr = \begin{bmatrix} r_1 \\ 2r_2 \\ 2r_3 \\ r_4 \end{bmatrix}$$

So, r_2 and r_3 are now ≈ 2 times more important, not 3, so they are more relaxed (higher than in b)

d) $$W = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & -1 & 0 \\ 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Wr = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & -1 & 0 \\ 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} r_1 \\ 3r_2-r_3 \\ 3r_3-r_2 \\ r_4 \end{bmatrix} = \begin{bmatrix} r_1 \\ 4r_2 \\ 4r_3 \\ r_4 \end{bmatrix} - \begin{bmatrix} 0 \\ r_2+r_3 \\ r_2+r_3 \\ r_4 \end{bmatrix}$$

$$\alpha^* = (A^T W^T W A)^{-1} A^T W^T W y$$

$$\boxed{\alpha^* = \begin{bmatrix} 0.08 \\ 1.68 \end{bmatrix}}$$

Plot - See Matlab

Residuals: $r_1 = 3.92$, $r_2 = -0.76$, $r_3 = 0.56$, $r_4 = -3.12$

Residual Norm Squared: $\|\vec{r}\|_2^2 = 25.9920$

The difference now is that r_2 and r_3 have decreased in magnitude, while r_1 and r_4 have increased. This suggests r_2 and r_3 are more important now. To show this:

$$\|Wr\|^2 = r_1^2 + (4r_2-(r_2+r_3))^2 + (4r_3-(r_2+r_3))^2 + r_4^2 = r_1^2 + 16r_2^2 - 8r_2(r_2+r_3) + (r_2+r_3)^2 + 16r_3^2 - 8r_3(r_2+r_3) + (r_2+r_3)^2 + r_4^2$$

$$= r_1^2 + 16r_2^2 - 6(r_2+r_3)^2 + 16r_3^2 + r_4^2$$

So, we want $r_i \approx 0 \ \forall i \in \{1,2,3,4\}$ and $r_2+r_3 \approx 0 \rightarrow r_2 \approx -r_3$ as before. This time, we get

$$Wr = \begin{bmatrix} r_1 \\ 4r_2 \\ 4r_3 \\ r_4 \end{bmatrix} - \begin{bmatrix} 0 \\ r_2+r_3 \\ r_2+r_3 \\ 0 \end{bmatrix} \approx \begin{bmatrix} r_1 \\ 4r_2 \\ 4r_3 \\ r_4 \end{bmatrix}$$

So, r_2 and r_3 are now ≈ 4 times more important, so they should be made even smaller in magnitude than b (and c)

e) $f_{quad}(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2$

Formulate our least-squares problems:

$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ 1 & x_4 & x_4^2 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 4 \\ 2 \end{bmatrix}$$

$A\alpha = y$

$\alpha^* = (A^TA)^{-1}A^Ty$

$\boxed{\alpha^* = \begin{bmatrix} 3.45 \\ -1.05 \\ 0.25 \end{bmatrix}}$ (Matlab)

Plot: See Matlab

Residuals: $\vec{r} = \begin{bmatrix} 0.55 \\ -1.65 \\ 1.65 \\ -0.55 \end{bmatrix}$

Residual norm squared: $\|\vec{r}\|_2^2 = 6.05$

$f_{cubic}(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3$

$$\begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 1 & x_2 & x_2^2 & x_2^3 \\ 1 & x_3 & x_3^2 & x_3^3 \\ 1 & x_4 & x_4^2 & x_4^3 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 4 \\ 2 \end{bmatrix}$$

$A\alpha = y$

$\alpha^* = (A^TA)^{-1}A^Ty$

$\boxed{\alpha^* = \begin{bmatrix} 4 \\ -9.6667 \\ 8.5000 \\ -1.8333 \end{bmatrix}}$ (Matlab)

Plot: See Matlab

Residuals: $\vec{r} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Residual norm squared $= \|\vec{r}\|_2^2 = 0$

As we increase the degree of the polynomial fit, the residual norm squared decreases because our fit better matches data → higher degree polynomial means more degrees of freedom and flexibility in best-fit curve, so leads to more accurate fits. In the case of f_{cubic}, we have 4 degrees of freedom and 4 data points, so there exists a cubic function that passes through all points, such that $\|\vec{r}\|_2^2 = 0$.

f) $f_{cubic,\gamma=0.05}(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + 0.05\|\alpha\|_2^2$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ \sqrt{0.05} & 0 & 0 & 0 \\ 0 & \sqrt{0.05} & 0 & 0 \\ 0 & 0 & \sqrt{0.05} & 0 \\ 0 & 0 & 0 & \sqrt{0.05} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 4 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \boxed{\alpha^* = \begin{bmatrix} 3.2787 \\ -4.0273 \\ 3.6308 \\ -0.8040 \end{bmatrix}}$$

$f_{cubic,\gamma=0.5}(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + 0.5\|\alpha\|_2^2$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \\ \sqrt{0.5} & 0 & 0 & 0 \\ 0 & \sqrt{0.5} & 0 & 0 \\ 0 & 0 & \sqrt{0.5} & 0 \\ 0 & 0 & 0 & \sqrt{0.5} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 4 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \rightarrow \boxed{\alpha^* = \begin{bmatrix} 2.7138 \\ -0.2946 \\ 0.2807 \\ -0.7157 \end{bmatrix}}$$

See Matlab code

5.3)

1) $x^a y = b$

$a\lg x + \lg y = \lg b$

$\lg b - a\lg x = \lg y$

$\begin{bmatrix} 1 & -\lg x \end{bmatrix} \begin{bmatrix} \lg b \\ a \end{bmatrix} = \lg y$

Stacking this for (x_i, y_i), $i = [m]$, we get:

$$\begin{bmatrix} 1 & -\lg x_1 \\ 1 & -\lg x_2 \\ \vdots & \vdots \\ 1 & -\lg x_m \end{bmatrix} \begin{bmatrix} \lg b \\ a \end{bmatrix} = \begin{bmatrix} \lg y_1 \\ \lg y_2 \\ \vdots \\ \lg y_m \end{bmatrix}$$

rewrite as: $Ax = y$

LS soln: $\boxed{A^T A x^* = A^T y}$

2) Unique soln if $A^T A$ is invertible.

Condition 1) $m \geq 2$

Condition 2) $\operatorname{rank}(A) = 2$ $\Rightarrow$ so that $\operatorname{rank}(A^T A) = \operatorname{rank}(A) = 2$, noting that $A^T A \in \mathbb{R}^{2\times 2}$ and thus would be invertible if $\operatorname{rank}(A) = 2$.

Condition 1 holds if we collect at least 2 data points

Condition 2 holds if not all authors have the same number of publications.

5.4)

a) $\gamma = 0 \rightarrow \min_{x \in \mathbb{R}^2} \|Ax - y\|_2^2$

Least squares solution: $A^T A x^* = A^T y$

$x^* = (A^T A)^{-1} A^T y$

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

$= \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix}$

$A^T y = \begin{bmatrix} 1 & 1 & 2 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 9 \\ -2 \\ 5 \end{bmatrix} = \begin{bmatrix} 12 \\ 11 \end{bmatrix}$

$x^* = \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix}^{-1} \begin{bmatrix} 12 \\ 11 \end{bmatrix} = \frac{1}{14} \begin{bmatrix} 3 & -2 \\ -2 & 6 \end{bmatrix} \begin{bmatrix} 12 \\ 11 \end{bmatrix} = \frac{1}{14} \begin{bmatrix} 14 \\ 42 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$

b) $Ax - y = A(x - x_0^* + x_0^*) - y$

$= A((x - x_0^*) + x_0^*) - y$

$= (Ax_0^* - y) + A(x - x_0^*)$

$Ax_0^* - y \in R(A)^\perp$, since Ax_0^* is projection of y onto A

$A(x - x_0^*) \in R(A)$

so, $(Ax_0^* - y) \perp (A(x - x_0^*))$, so we can apply Pythagorean theorem:

$\|Ax - y\|_2^2 = \|Ax_0^* - y\|_2^2 + \|A(x - x_0^*)\|_2^2$, as desired

c) In the case where $\gamma = 0$, objective of (2) reduces to $\|A(x - x_0^*)\|_2^2 + \|Ax_0^* - y\|_2^2$. Since $Ax_0^* - y$ is known, we only focus on $\|A(x - x_0^*)\|_2^2$.

Say $\|A(x - x_0^*)\|_2^2 = c$, where $c \geq \|Ax_0^* - y\|_2^2$

$\Rightarrow (x - x_0^*)^T A^T A (x - x_0^*) = c$

$\tilde{x}^T A^T A \tilde{x} = c$

$A^T A = \begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix}$

$\det(A^T A - \lambda I) = \det\left(\begin{bmatrix} 6-\lambda & 2 \\ 2 & 3-\lambda \end{bmatrix}\right) = (\lambda - 6)(\lambda - 3) - 4$

$= \lambda^2 - 9\lambda + 14 = (\lambda - 7)(\lambda - 2)$

$\begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$ $\quad 6x + 2y = 2x \rightarrow 2y = 4x \rightarrow y = -2x$

$v_1 = \begin{bmatrix} 1 \\ -2 \end{bmatrix} / \sqrt{5}$

$\begin{bmatrix} 6 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7x \\ 7y \end{bmatrix}$ $\quad 6x + 2y = 7x \rightarrow x = 2y$

$v_2 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} / \sqrt{5}$

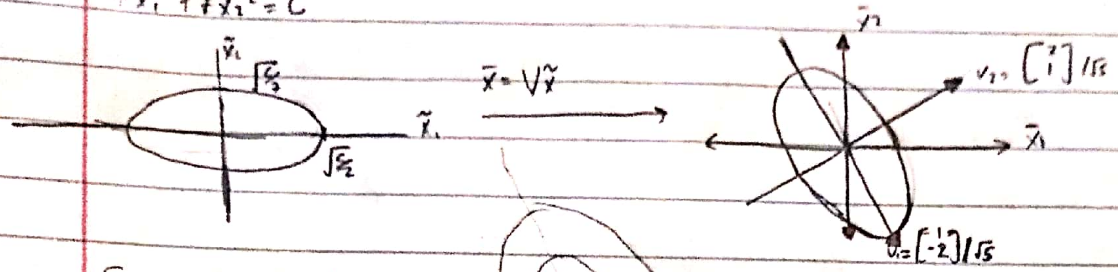
$A^T A = V \Lambda V^T$

$\tilde{x}^T V \Lambda V^T \tilde{x} = c$

$\tilde{x}^T \Lambda \tilde{x} = c$

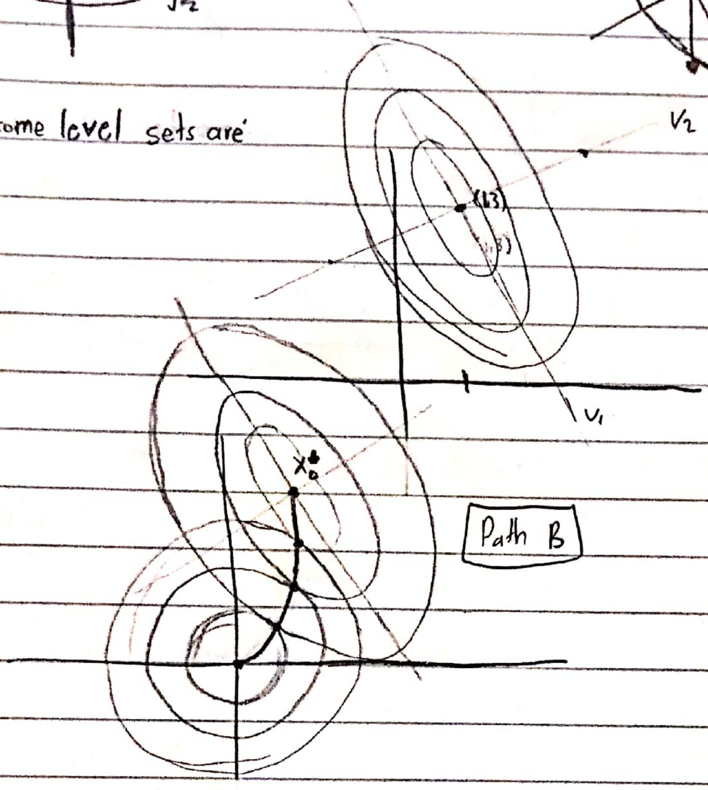
$\tilde{x}^T \begin{bmatrix} 2 & 0 \\ 0 & 7 \end{bmatrix} \tilde{x} = c$

$2\tilde{x}_1^2 + 7\tilde{x}_2^2 = c$



So, some level sets are:

d)



5.5a) $A^TAx^* = A^TA(A^+y) = A^TA(V_r\Sigma^{-1}U_r^Ty)$

$A = U\tilde{\Sigma}V^T$

$A^TA = V\tilde{\Sigma}^TU^TU\tilde{\Sigma}V^T = V\tilde{\Sigma}^2V^T = V_r\Sigma^2V_r^T$

So: $A^TAx^* = V_r\Sigma^2V_r^TV_r\Sigma^{-1}U_r^Ty = V_r\Sigma^2\Sigma^{-1}U_r^Ty = V_r\Sigma U_r^Ty = Ay$ ✓

b)

i) $x^* = A^+y = V_r\Sigma^{-1}U_r^Ty$

$A = U\tilde{\Sigma}V^T = U_r\Sigma V_r^T$ $\qquad U = A\Sigma B^T$

$A^T = V\tilde{\Sigma}U^T = V_r\Sigma U_r^T$

$x^* = V_r\Sigma^{-1}U_r^Ty = V_r(\Sigma \underbrace{U_r^TU_r}_{}\Sigma^{-2})U_r^Ty = (V_r\Sigma U_r^T)(U_r\Sigma^{-2}U_r^Ty) = A^T(U_r\Sigma^{-2}U_r^Ty) \in R(A^T)$ $\qquad = \Sigma^{-1}$

ii) $Ax^* = y$

$\forall y \in \mathbb{R}^m$

$A \in \mathbb{R}^{m \times n}$ $\quad x^* \in \mathbb{R}^{n \times 1}$

To span all of $\mathbb{R}^m$, we must have rank $A \geq m$, ie $\geq m$ linearly independent columns. Since $\text{rank}(A) \leq \min(m,n)$, this means $\boxed{\text{rank } A = m}$. If rank $A = m$, m lin. ind. columns means we can span all of $\mathbb{R}^m$, in atleast one way.

c) $x^* = A^\dagger y = V_r \Sigma^{-1} U_r^T y$

$A x^* = U_r \Sigma V_r^T V_r \Sigma^{-1} U_r^T y = U_r U_r^T y = \sum_{i=1}^{r} \langle y, u^{(i)} \rangle u^{(i)}$ (1)

Now, note that $R(A)$ = span of columns of U_r, ie columns of U_r form an orthonormal basis for $R(A)$.
The projection of y onto $R(A)$ is thus:

$y_s = \alpha_1 u^{(1)} + \dots \alpha_r u^{(r)}$

$\langle y, u^{(i)} \rangle = \langle y_s + y_{s\perp}, u^{(i)} \rangle = \langle y_s, u^{(i)} \rangle + \langle y_{s\perp}, u^{(i)} \rangle = \alpha_i + 0 = \alpha_i$

So, $\alpha_i = \langle y, u^{(i)} \rangle$ and so

$\Pi_{R(A)} y = \sum_{i=1}^{r} \langle y, u^{(i)} \rangle u^{(i)}$ - (2)

Combining (1) + (2) gives $A x^* = \Pi_{R(A)} y$, ie $\boxed{A x^* \text{ is projection of } y \text{ onto } R(A)}$

From b), ii, $\boxed{\|x^*\|_2 \text{ is the minimum } x^* \text{ st } A x^* = \Pi_{R(A)} y}$

Scanned with CamScanner

5.7)

a) $\dot{x}(n) = \dot{x}(n-1) + \int_{n-1}^{n} \ddot{x}(t)\,dt$

$= \dot{x}(n-1) + \int_{n-1}^{n} \frac{f(t)}{m}\,dt$

$m=1$, $f(t) = p_n$ for $n-1 \le t < n$:

$\therefore \dot{x}(n) = \dot{x}(n-1) + \int_{n-1}^{n} p_n\,dt$

$= \dot{x}(n-1) + p_n \int_{n-1}^{n} dt$

$\boxed{\dot{x}(n) = \dot{x}(n-1) + p_n}$

$x(n) = x(n-1) + \int_{n-1}^{n} \dot{x}(t)\,dt$

$\dot{x}(t) = \dot{x}(n-1) + \int_{n-1}^{t} p_n\,dt = \dot{x}(n-1) + p_n(t-n+1)$

$x(n) = x(n-1) + \int_{n-1}^{n} \dot{x}(n-1) + p_n(t-n+1)\,dt$

$= x(n-1) + \int_{n-1}^{n} \dot{x}(n-1)\,dt + p_n \int_{n-1}^{n} (t-n+1)\,dt$

$= x(n-1) + \dot{x}(n-1) + p_n \left. \frac{(t-n+1)^2}{2} \right|_{n-1}^{n}$

$= x(n-1) + \dot{x}(n-1) + p_n \left[\frac{(n-n+1)^2}{2} - \frac{(n-1-n+1)^2}{2} \right]$

$= x(n-1) + \dot{x}(n-1) + p_n\left(\frac{1}{2} - 0\right)$

$\boxed{x(n) = x(n-1) + \dot{x}(n-1) + \frac{1}{2} p_n}$

b) $x(0) = 0$ $\quad \dot{x}(0) = 0$ $\quad x(10) = 1$ $\quad \dot{x}(10) = 0$

$\vec{x}_n = A\vec{x}_{n-1} + bp_n$

$= A(A\vec{x}_{n-2} + bp_{n-1}) + bp_n$

$= A^2\vec{x}_{n-2} + Abp_{n-1} + bp_n$

$= A^n \vec{x}_0 + A^{n-1}bp_1 + A^{n-2}bp_2 + \dots + Abp_{n-1} + bp_n$

$\vec{x}_n = A^{n-1}bp_1 + A^{n-2}bp_2 + \dots + bp_n$

$$\begin{bmatrix} b & 0 & \dots & \dots & 0 \\ Ab & b & 0 & \dots & 0 \\ A^2b & Ab & b & \dots & \\ & & & & \\ A^9b & \dots & \dots & \dots & b \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ \vdots \\ p_{10} \end{bmatrix} = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vdots \\ \vec{x}_{10} \end{bmatrix}$$

$\vec{x}_1 = bp_1$

$\vec{x}_2 = Abp_1 + bp_2$

$\vec{x}_3 = A^2bp_1 + Abp_2 + bp_3$

$$\underbrace{\begin{bmatrix} A^9b & A^8b & \dots & A^0b \end{bmatrix}}_{\tilde{A}} \underbrace{\begin{bmatrix} p_1 \\ \vdots \\ p_{10} \end{bmatrix}}_{p} = \underbrace{\vec{x}_{10}}_{\tilde{y}}$$

(Only constraint, the other constraint was used to get here)

$\tilde{A}x = \tilde{y}$

Underdetermined LS problem: $p^* = \tilde{A}^{+}\tilde{y} \Rightarrow$

$$\boxed{p^* = \begin{bmatrix} 0.0545 \\ 0.0424 \\ 0.0303 \\ 0.0182 \\ 0.0061 \\ -0.0061 \\ -0.0182 \\ -0.0303 \\ -0.0424 \\ -0.0545 \end{bmatrix}}$$ (Matlab)

See plots in Matlab code

$\dot{x}(t) = \dot{x}(\lfloor t \rfloor) + p_n(t - \lfloor t \rfloor)$

$= \text{vel}(\lfloor t \rfloor) + p_{ls}(\lfloor t \rfloor + 1)(t - \lfloor t \rfloor)$

$\dot{x}(7.5) = \dot{x}(7) + p(2) \times 0.5$

$x(t) = x(\lfloor t \rfloor) + \int_{\lfloor t \rfloor}^{t} \dot{x}(\lfloor t \rfloor)\,dt + p(\lfloor t \rfloor) \int_{\lfloor t \rfloor}^{t} (t - \lfloor t \rfloor)\,dt = x(\lfloor t \rfloor) + \dot{x}(\lfloor t \rfloor)(t - \lfloor t \rfloor) + p(\lfloor t \rfloor)\left[\int_{\lfloor t \rfloor}^{t} t - \lfloor t \rfloor (t - \lfloor t \rfloor) \right]$

$= x(\lfloor t \rfloor) + \dot{x}(\lfloor t \rfloor)(t - \lfloor t \rfloor) + p(\lfloor t \rfloor)\left[\frac{1}{2}(t^2 - \lfloor t \rfloor^2) - \lfloor t \rfloor (t - \lfloor t \rfloor) \right]$

Scanned with CamScanner

Intuition:

Want to get from $x(0)=0, x'(0)=0$ to $x(10)=1$, $x'(10)=0$, with minimum ?: measure of force.
To do this, accelerate block for first 5 seconds to halfway point, then decelerate for next 5 second so we reach $x=10$ at velocity 0.

c) This one requires one additional step

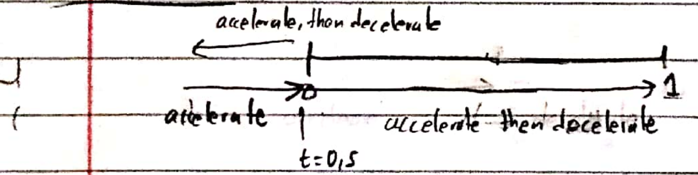
$$\begin{bmatrix} A^9b & A^8b & A^7b & A^6b & A^5b & A^4b & A^3b & A^2b & Ab & b \\ A^4b & A^3b & A^2b & Ab & b & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_{10} \end{bmatrix} = \begin{bmatrix} \vec{x}_{10} \\ \vec{x}_5 \end{bmatrix}$$

Since $\begin{bmatrix} \vec{x}_{10} \\ \vec{x}_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \text{anything} \end{bmatrix}$, we just remove the last row of leftmost matrix. Call this $\tilde{A}$ and let $\tilde{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$. Then:

$$\tilde{A}p = \tilde{y}$$

$$p = \tilde{A}^{+}\tilde{y} \Rightarrow \boxed{p^{*} = \begin{bmatrix} -0.0955 \\ -0.0076 \\ 0.0303 \\ 0.0682 \\ 0.1061 \\ 0.0939 \\ 0.0318 \\ -0.0303 \\ -0.0924 \\ -0.1545 \end{bmatrix} \text{(Matlab)}}$$

Intuition:



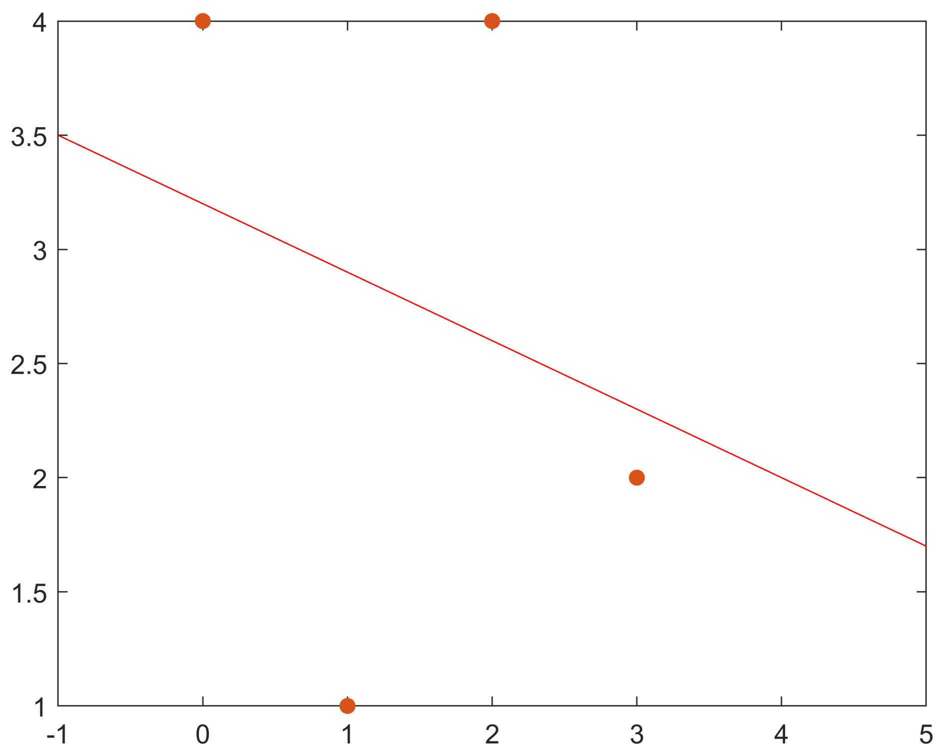
We move block back, then we accelerate it. Now @ t=5, it is at 0 but with some non zero velocity, so it is easier for block to reach end position $x(10)=1$.
If we instead waited till t=5 to do anything, we would have 0 velocity to start with and the extra effort to get block going is more than the effort to get block starting with non-zero velocity (arrows A and B above accomplish this latter part)

Problem 5.2a

```
A = [1,0;1,1;1,2;1,3];
x = [0;1;2;3];
y = [4;1;4;2];

alpha = (A' * A)\(A' * y);

xvals = -1:0.01:5;
yvals = alpha(1) + alpha(2) * xvals;
plot(xvals, yvals, 'red')
hold on
scatter(x,y, 'filled')
```




```
yls = alpha(1) + alpha(2) * x;
r = y - yls
```

```
r = 4×1
    0.8000
   -1.9000
    1.4000
   -0.3000
```

```
residual_norm_squared = norm(r)^2
```

```
residual_norm_squared = 6.3000
```

Problem 5.2b

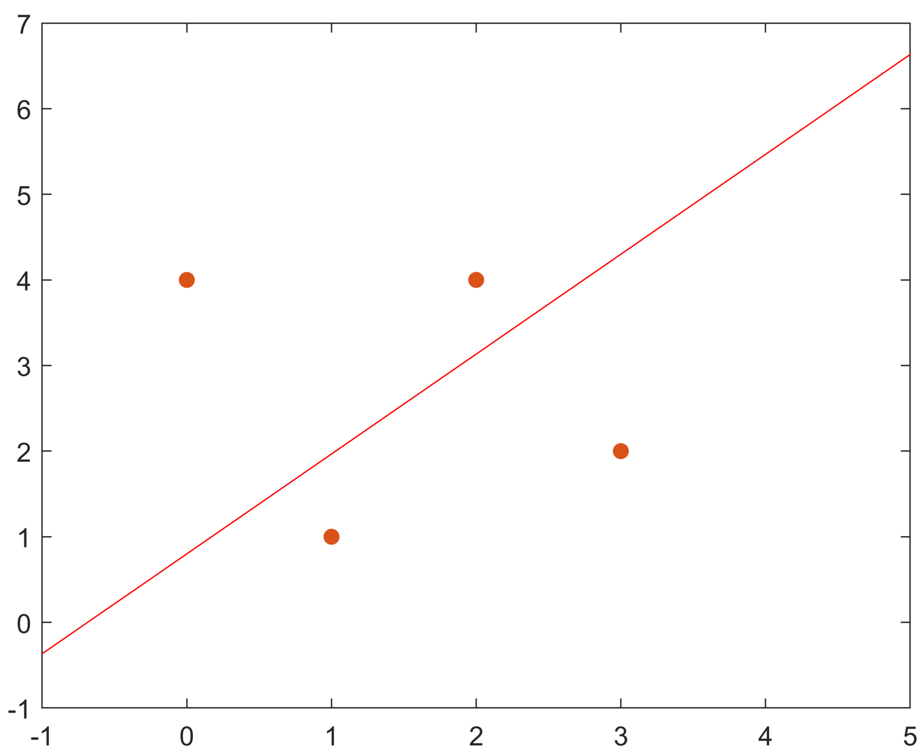
```
A = [1,0;1,1;1,2;1,3];
x = [0;1;2;3];
y = [4;1;4;2];
W = diag([1,3,3,1])
```

```
W = 4×4
     1     0     0     0
     0     3     0     0
     0     0     3     0
     0     0     0     1
```

```
alpha = (A' * (W' * W) * A)\(A' * (W' * W) * y)
```

```
alpha = 2×1
    0.8000
    1.1667
```

```
xvals = -1:0.01:5;
yvals = alpha(1) + alpha(2) * xvals;
figure;
plot(xvals, yvals, 'red')
hold on
scatter(x,y, 'filled')
```




```
yls = alpha(1) + alpha(2) * x;
r = y - yls
```

```
r = 4×1
    3.2000
   -0.9667
```

```
    0.8667
   -2.3000
```

```
residual_norm_squared = norm(r)^2
```

```
residual_norm_squared = 17.2156
```

Problem 5.2c

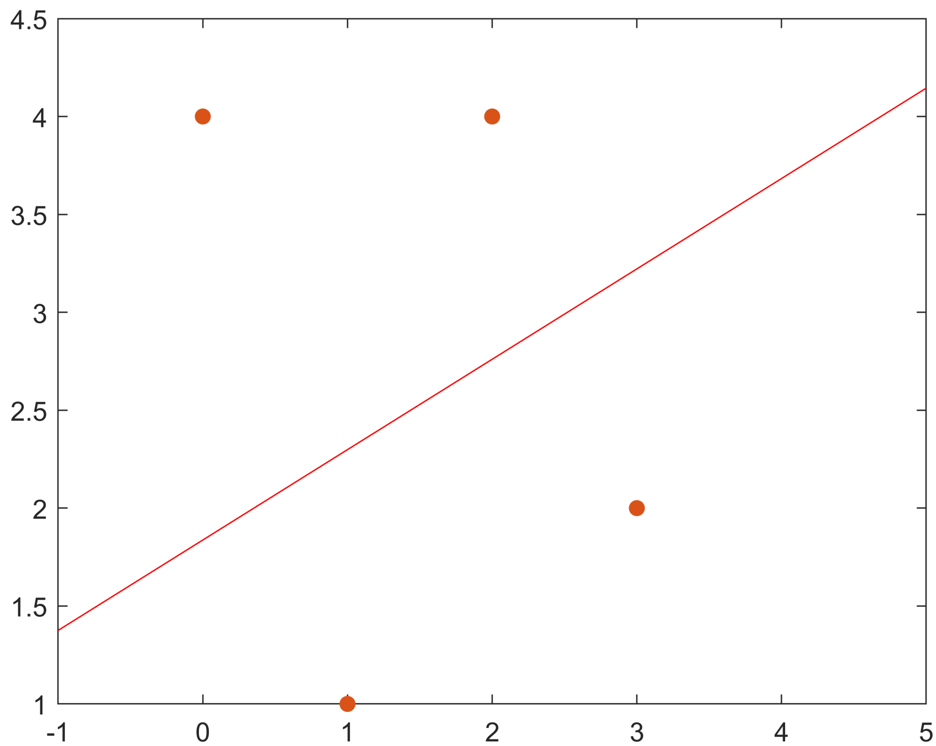
```
A = [1,0;1,1;1,2;1,3];
x = [0;1;2;3];
y = [4;1;4;2];
W = [1,0,0,0;0,3,1,0;0,1,3,0;0,0,0,1]
```

```
W = 4×4
     1     0     0     0
     0     3     1     0
     0     1     3     0
     0     0     0     1
```

```
alpha = (A' * (W' * W) * A)\(A' * (W' * W) * y)
```

```
alpha = 2×1
    1.8371
    0.4615
```

```
xvals = -1:0.01:5;
yvals = alpha(1) + alpha(2) * xvals;
figure;
plot(xvals, yvals, 'red')
hold on
scatter(x,y, 'filled')
```




```
yls = alpha(1) + alpha(2) * x;
r = y - yls
```

```
r = 4×1
    2.1629
   -1.2986
    1.2398
   -1.2217
```

```
residual_norm_squared = norm(r)^2
```

```
residual_norm_squared = 9.3943
```

Problem 5.2d

```
A = [1,0;1,1;1,2;1,3];
x = [0;1;2;3];
y = [4;1;4;2];
W = [1,0,0,0;0,3,-1,0;0,-1,3,0;0,0,0,1]
```

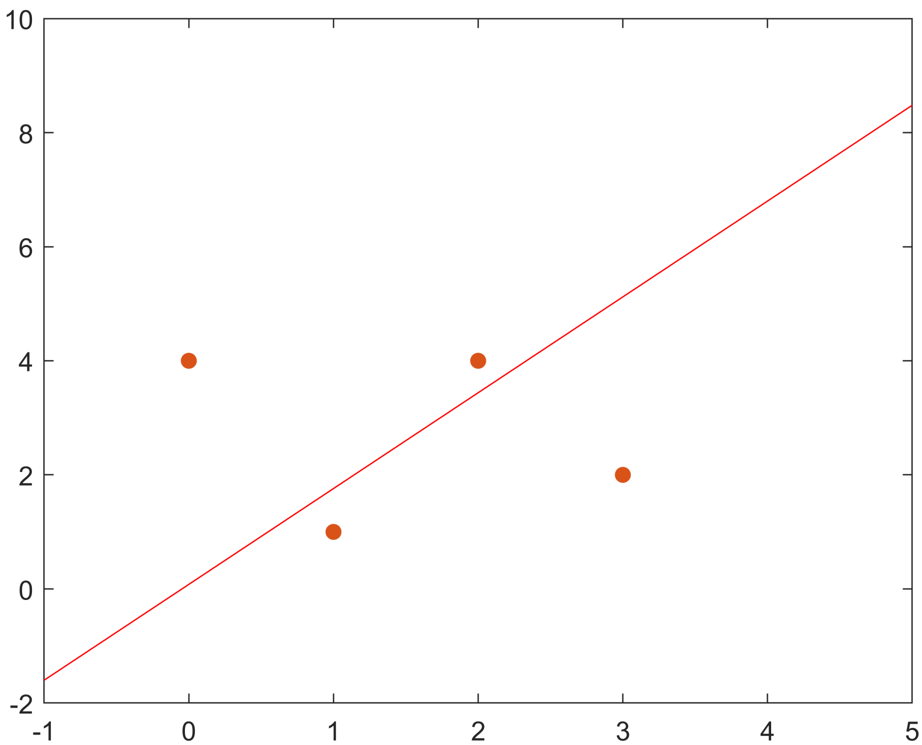
```
W = 4×4
     1     0     0     0
     0     3    -1     0
     0    -1     3     0
     0     0     0     1
```

```
alpha = (A' * (W' * W) * A)\(A' * (W' * W) * y)
```

```
alpha = 2×1
    0.0800
```

1.6800

```
xvals = -1:0.01:5;
yvals = alpha(1) + alpha(2) * xvals;
figure;
plot(xvals, yvals, 'red')
hold on
scatter(x,y, 'filled')
```




```
yls = alpha(1) + alpha(2) * x;
r = y - yls
```

```
r = 4×1
    3.9200
   -0.7600
    0.5600
   -3.1200
```

```
residual_norm_squared = norm(r)^2
```

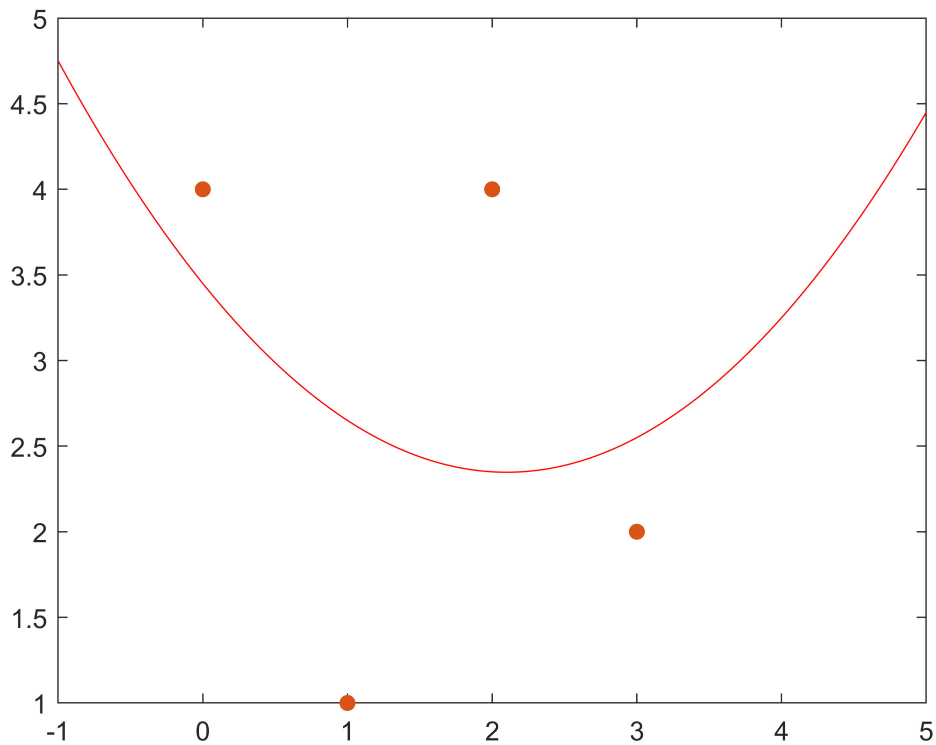
```
residual_norm_squared = 25.9920
```

Problem 5.2e

```
A = [1,0,0;1,1,1;1,2,4;1,3,9];
x = [0;1;2;3];
y = [4;1;4;2];
alpha = (A' * A)\(A' * y)
```

```
alpha = 3×1
    3.4500
   -1.0500
    0.2500
```

```
xvals = -1:0.01:5;
yvals = alpha(1) + alpha(2) * xvals + alpha(3) * xvals.^2;
figure;
plot(xvals, yvals, 'red')
hold on
scatter(x,y, 'filled')
```




```
yls = alpha(1) + alpha(2) * x + alpha(3) * x.^2;
r = y - yls
```

```
r = 4×1
    0.5500
   -1.6500
    1.6500
   -0.5500
```

```
residual_norm_squared = norm(r)^2
```

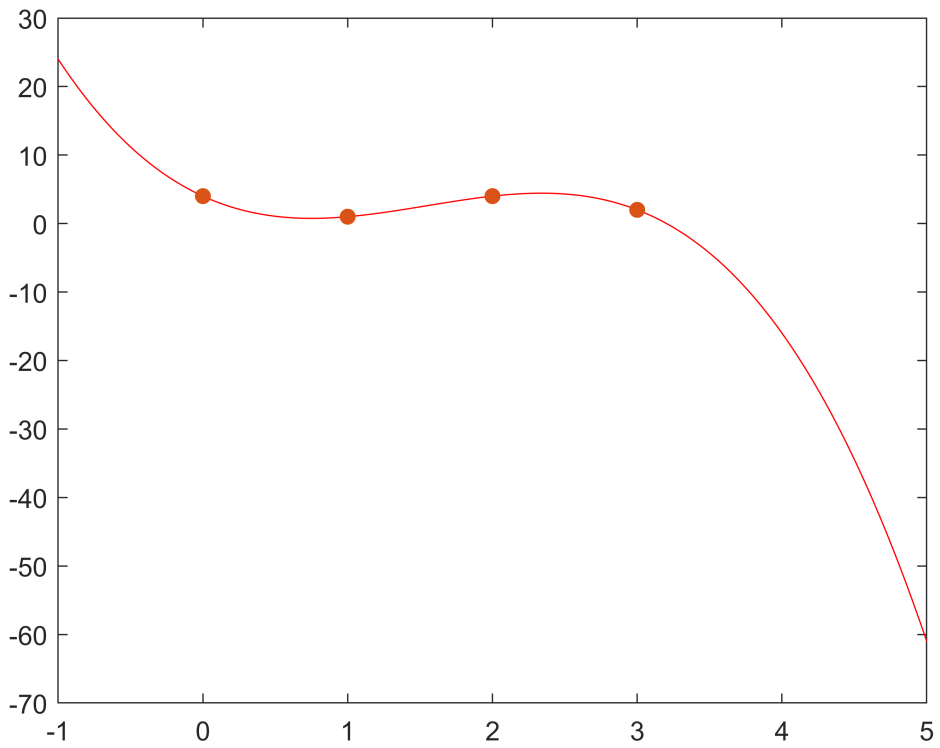
```
residual_norm_squared = 6.0500
```

```
Acubic = [1,0,0,0;1,1,1,1;1,2,4,8;1,3,9,27];
alphacubic = (Acubic' * Acubic)\(Acubic' * y)
```

```
alphacubic = 4×1
    4.0000
   -9.6667
```

```
    8.5000
   -1.8333
```

```
xvals = -1:0.01:5;
yvalscubic = alphacubic(1) + alphacubic(2) * xvals + alphacubic(3) * xvals.^2 + alphacubic(4) *
figure;
plot(xvals, yvalscubic, 'red')
hold on
scatter(x,y, 'filled')
```




```
ylscubic = alphacubic(1) + alphacubic(2) * x + alphacubic(3) * x.^2 + alphacubic(4) * x.^3;
rcubic = y - ylscubic
```

```
rcubic = 4×1
10^-13 ×
    0.2132
   -0.4508
    0.3020
   -0.0711
```

```
cubicresidual_norm_squared = norm(rcubic)^2
```

```
cubicresidual_norm_squared = 3.4486e-27
```

Problem 5.2f

```
A0 = [1,0,0,0;1,1,1,1;1,2,4,8;1,3,9,27;0,0,0,0;0,0,0,0;0,0,0,0;0,0,0,0];
A1 = [1,0,0,0;1,1,1,1;1,2,4,8;1,3,9,27;sqrt(0.05),0,0,0;0,sqrt(0.05),0,0;0,0,sqrt(0.05),0;0,0,0
A2 = [1,0,0,0;1,1,1,1;1,2,4,8;1,3,9,27;sqrt(0.5),0,0,0;0,sqrt(0.5),0,0;0,0,sqrt(0.5),0;0,0,0,sq
```

```
y = [4;1;4;2;0;0;0;0];
alpha0 = (A0' * A0)\(A0' * y)
```

```
alpha0 = 4×1
    4.0000
   -9.6667
    8.5000
   -1.8333
```

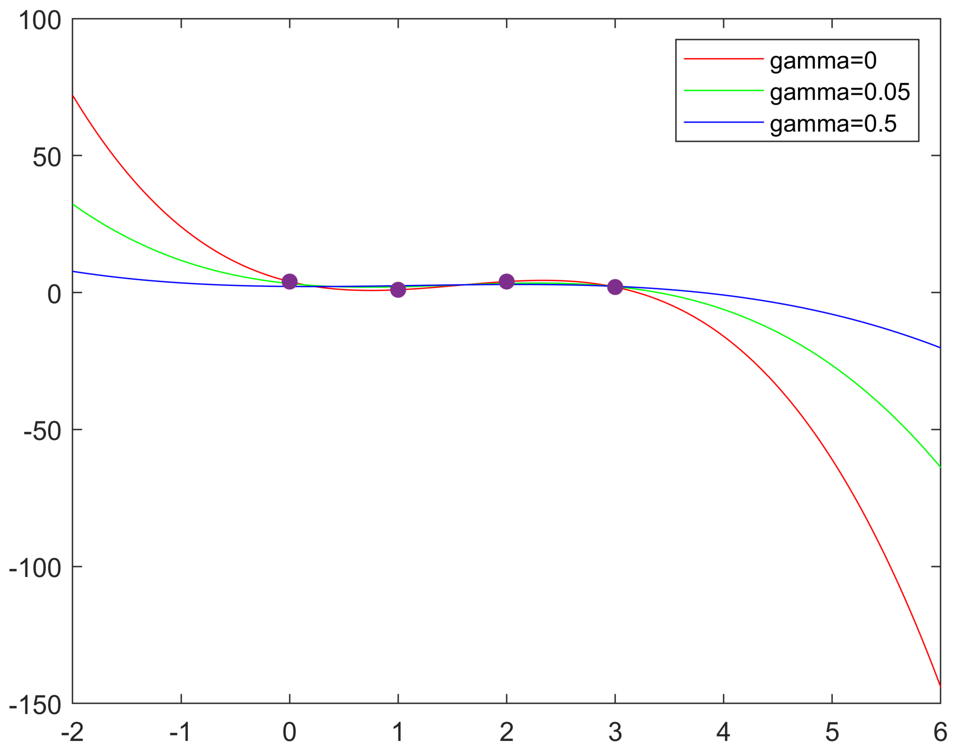
```
alpha1 = (A1' * A1)\(A1' * y)
```

```
alpha1 = 4×1
    3.2787
   -4.0273
    3.6308
   -0.8040
```

```
alpha2 = (A2' * A2)\(A2' * y)
```

```
alpha2 = 4×1
    2.2138
   -0.2946
    0.7807
   -0.2257
```

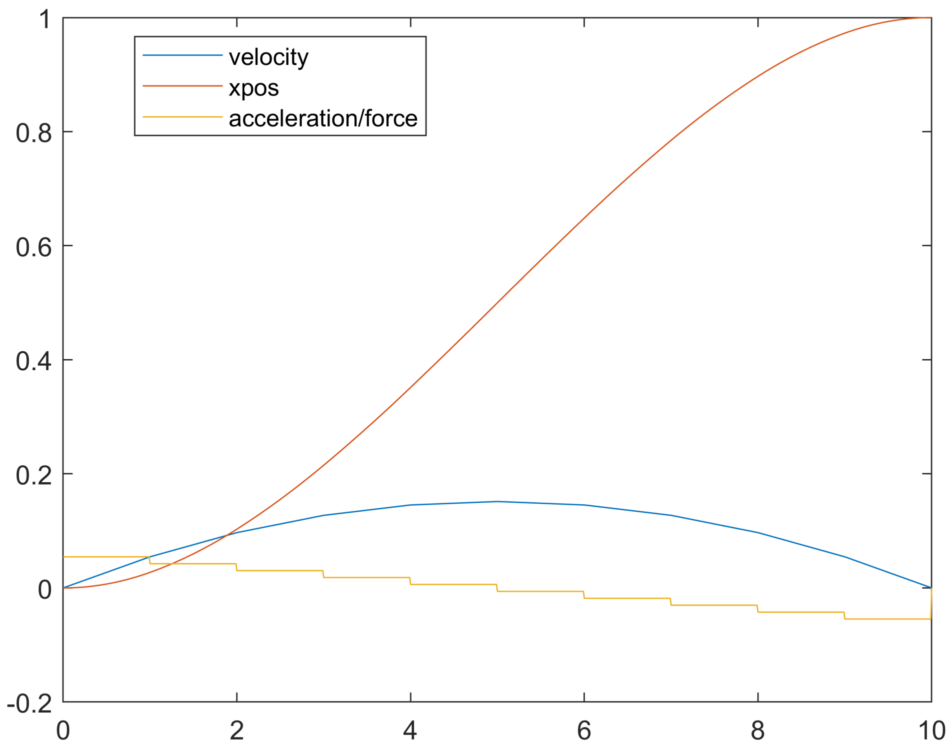
```
xvals = -2:0.01:6;
yvals0 = alpha0(1) + alpha0(2) * xvals + alpha0(3) * xvals.^2 + alpha0(4) * xvals.^3;
yvals1 = alpha1(1) + alpha1(2) * xvals + alpha1(3) * xvals.^2 + alpha1(4) * xvals.^3;
yvals2 = alpha2(1) + alpha2(2) * xvals + alpha2(3) * xvals.^2 + alpha2(4) * xvals.^3;
figure;
plot(xvals, yvals0, 'red')
hold on
plot(xvals, yvals1, 'green')
hold on
plot(xvals, yvals2, 'blue')
hold on
scatter(x,y(1:4), 'filled')
legend('gamma=0', 'gamma=0.05', 'gamma=0.5')
```



Problem 5.7b

```
A = [1,1;0,1];
b = [0.5;1];
Als = [A^9*b A^8*b A^7*b A^6*b A^5*b A^4*b A^3*b A^2*b A*b b];
yls = [1;0];
pls = pinv(Als)*yls;
norm(pls);
pos = [0];
vel = [0];
for i=1:10
    step = A*[pos(end); vel(end)] + pls(i)*b;
    pos = [pos step(1)];
    vel = [vel step(2)];
end
pos;
vel;
timesteps = 0:0.01:10;
pn = [pls; 0];
f = @(t) pn(floor(t)+1);
xbar = @(t) vel(floor(t)+1) + pn(floor(t)+1) * (t - floor(t));
x = @(t) pos(floor(t)+1) + vel(floor(t)+1) * (t-floor(t)) + pn(floor(t)+1) * (0.5*(t.^2-(floor(
%
figure;
plot(timesteps, arrayfun(xbar, timesteps))
hold on
plot(timesteps, arrayfun(x, timesteps))
```

```
hold on
plot(timesteps, arrayfun(f, timesteps))
legend('velocity', 'xpos', 'acceleration/force', 'Location', 'Best')
```



Problem 5.7c

```
A = [1,1;0,1];
b = [0.5;1];
Als1 = [A^9*b A^8*b A^7*b A^6*b A^5*b A^4*b A^3*b A^2*b A*b b];
Als2tmp = [A^4*b A^3*b A^2*b A*b b];
z = [0;0];
Als2 = [Als2tmp z z z z z];
Als = [Als1; Als2];
Als = Als(1:3, :)
```

```
Als = 3×10
    9.5000    8.5000    7.5000    6.5000    5.5000    4.5000    3.5000    2.5000 ···
    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000    1.0000
    4.5000    3.5000    2.5000    1.5000    0.5000         0         0         0
```

```
yls = [1;0;0]
```

```
yls = 3×1
     1
     0
     0
```

```
pls = pinv(Als)*yls
```

```
pls = 10×1
```

```
   -0.0455
   -0.0076
    0.0303
    0.0682
    0.1061
    0.0939
    0.0318
   -0.0303
   -0.0924
   -0.1545
```

```
norm(pls)
```

```
ans = 0.2492
```

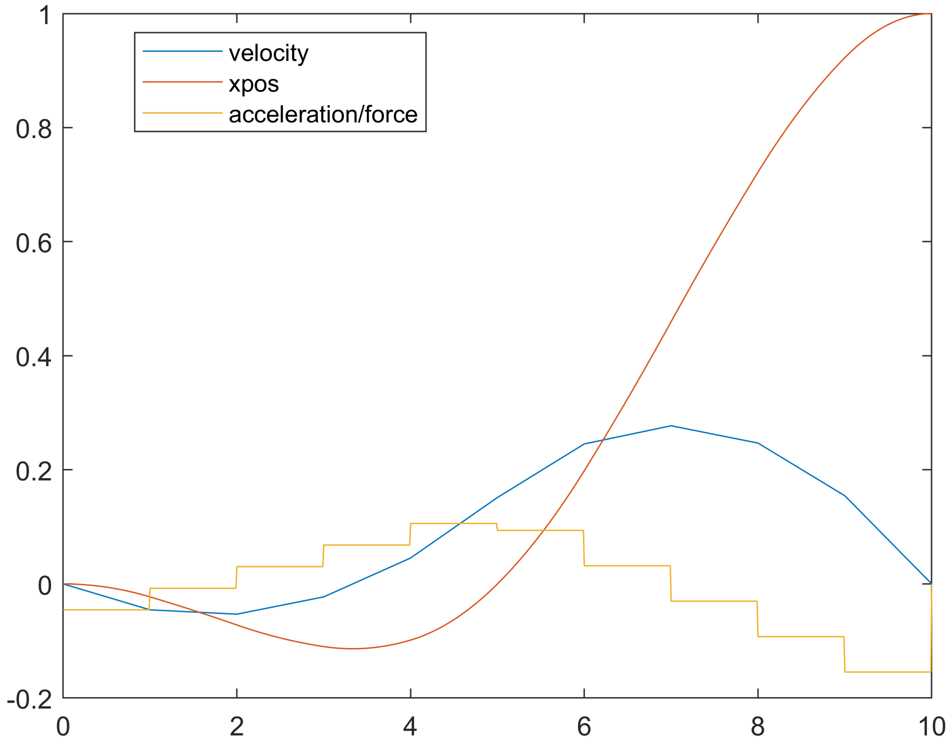
```
pos = [0];
vel = [0];
for i=1:10
    step = A*[pos(end); vel(end)] + pls(i)*b;
    pos = [pos step(1)];
    vel = [vel step(2)];
end
pos
```

```
pos = 1×11
         0   -0.0227   -0.0720   -0.1098   -0.0985    0.0000    0.1985    0.4598 ...
```

```
vel
```

```
vel = 1×11
         0   -0.0455   -0.0530   -0.0227    0.0455    0.1515    0.2455    0.2773 ...
```

```
timesteps = 0:0.01:10;
pn = [pls; 0];
f = @(t) pn(floor(t)+1);
xbar = @(t) vel(floor(t)+1) + pn(floor(t)+1) * (t - floor(t));
x = @(t) pos(floor(t)+1) + vel(floor(t)+1) * (t-floor(t)) + pn(floor(t)+1) * (0.5*(t.^2-(floor(
%
figure;
plot(timesteps, arrayfun(xbar, timesteps))
hold on
plot(timesteps, arrayfun(x, timesteps))
hold on
plot(timesteps, arrayfun(f, timesteps))
legend('velocity', 'xpos', 'acceleration/force', 'Location', 'Best')
```


velocity
xpos
acceleration/force
1
0.8
0.6
0.4
0.2
0
-0.2
0
2
4
6
8
10