1.

a) Since $E(g(x)) \in \partial f(x)$, we have: $f(x_k) + E[g(x_k)]^T (x^* - x_k) \le f(x^*)$

So, $\underbrace{E[g(x_k)]^T (x^* - x_k)}$ $\le f(x^*) - f(x_k)$ $\qquad E[\|g(x)\|^2] \le M^2$

$$\|x_{k+1} - x^*\|^2 = \|x_k - a_k g(x_k) - x^*\|^2 = \|x_k - x^*\|^2 - 2\|x_k - x^*\|^T \cdot a_k \cdot g(x_k) + a_k^2 \|g(x_k)\|^2$$

$$\|x_k - x^*\|^2 - 2(x_k - x^*)^T \cdot a_k \cdot g(x_k) + a_k^2 \|g(x_k)\|^2 \le \|x_k - x^*\|^2 + 2a_k\left(f(x^*) - f(x_k)\right) + a_k^2 M^2$$

$$= \|x_k - x^*\|^2 - 2a_k\left(f(x_k) - f(x^*)\right) + a_k^2 M^2$$

so $E[\|x_{k+1} - x_k\|^2 | x_k] \le \|x_k - x^*\|^2 - 2a_k(f(x_k) - f(x^*)) + a_k^2 M^2$ $\qquad$ what we want.

b) from a) we have $E[\|x_{k+1} - x_k\|^2 | x_k] \le \|x_k - x^*\|^2 - 2a_k(f(x_k) - f(x^*)) + a_k^2 M^2$

so $2a_k(f(x_k) - f(x^*)) \le \|x_k - x^*\|^2 - E[\|x_{k+1} - x_k\|^2 | x_k] + a_k^2 M^2$

use expectation on x_k:

$$E\left[2a_k(f(x_k) - f(x^*))\right] \le E\left[\|x_k - x^*\|^2\right] - E\left[\|x_{k+1} - x_k\|^2\right] + a_k^2 M^2$$

$$\vdots$$

$$E\left[2a_0(f(x_0) - f(x^*))\right] \le E\left[\|x_0 - x^*\|^2\right] - E\left[\|x_1 - x_0\|^2\right] + a_0^2 M^2$$

so $$\sum_{i=0}^{k} 2a_i \cdot E[f(x_k) - f(x^*)] \le \|x_0 - x^*\|^2 - E\left[\|x_{k+1} - x_k\|^2\right] + \sum_{i=0}^{k} a_i^2 M^2$$

so $$\sum_{i=0}^{k} 2a_i \cdot E\left[\min_{i \le k}\{f(x_i) - f(x^*)\}\right] \le \|x_0 - x^*\|^2 + \sum_{i=0}^{k} a_i^2 M^2$$

so $$E\left(\min_{i \le k}\{f(x_i) - f(x^*)\}\right] \le \frac{\|x_0 - x^*\|^2 + M^2 \sum_{i=0}^{k} a_i^2}{\sum_{i=0}^{k} 2a_i}$$ $\leftarrow$ this is an upperbound

c) let $a_i = \frac{1}{\sqrt{k+1}}$

then, we have:

$$E\left(\min_{i \le k}\{f(x_i) - f(x^*)\}\right] \le \frac{\|x_0 - x^*\|^2 + M^2 \cdot \frac{k+1}{k+1}}{\frac{2k+2}{\sqrt{k+1}}} = \frac{\|x_0 - x^*\|^2 + M^2}{2\sqrt{k+1}}$$

So after k steps, the bound is at most $O(1/\sqrt{k})$

d) let $a_i = \frac{1}{\sqrt{i+1}}$

since $$E\left(\min_{i \le k}\{f(x_i) - f(x^*)\}\right] \le \frac{\|x_0 - x^*\|^2 + M^2 \sum_{i=0}^{k} a_i^2}{\sum_{i=0}^{k} 2a_i}$$

and $$\sum_{i=0}^{k} a_i = 1 + \frac{1}{\sqrt{2}} + \dots \frac{1}{\sqrt{k+1}} \le \int_{t\to 0}^{k+1} \frac{1}{\sqrt{x}} dx = 2\sqrt{x}\Big|_{t\to 0}^{k+1} = 2\sqrt{k+1}$$

$$\sum_{i=0}^{k} a_i^2 = \sum_{i=1}^{k+1} \frac{1}{i} \le \log(k+1)$$

so $$E\left(\min_{i \le k}\{f(x_i) - f(x^*)\}\right] \le \frac{\|x_0 - x^*\|^2 + M^2 \log(k+1)}{4\sqrt{k+1}} \sim O\left(\frac{\log k}{\sqrt{k}}\right)$$

2) a) $\nabla h(x) = \nabla\left(\frac{1}{2} F(x)^T F(x)\right) = \nabla F(x)^T \cdot F(x)$

$$\|\nabla h(x) - \nabla h(y)\| = \|\nabla F(x)^T F(x) - \nabla F(y)^T F(y)\|$$
$$= \|\nabla F(x)^T F(x) - \nabla F(x)^T F(y) + \nabla F(x)^T F(y) - \nabla F(y)^T F(y)\|$$
$$\le \|\nabla F(x)^T F(x) - \nabla F(x)^T F(y)\| + \|\nabla F(x)^T F(y) - \nabla F(y)^T F(y)\|$$

$$\|\nabla F(x)^T F(x) - \nabla F(x)^T F(y)\| \le \|\nabla F(x)\| \cdot \|F(x) - F(y)\| \le N \cdot L \cdot \|x - y\|$$

$$\|\nabla F(x)^T F(y) - \nabla F(y)^T F(y)\| \le \|\nabla F(x) - \nabla F(y)\| \cdot \|F(y)\| \le Q \cdot M \cdot \|x - y\|$$

so $\|\nabla h(x) - \nabla h(y)\| \le (NL + QM) \cdot \|x - y\|$

b) let $x_{k+1} = x_k - \alpha_k \nabla h(x_k)$

to get $\nabla h(x_k)$ we need:

① compute $\nabla F(x)$ .... $O(d^2)$

② compute $F(x)$:....$O(d)$

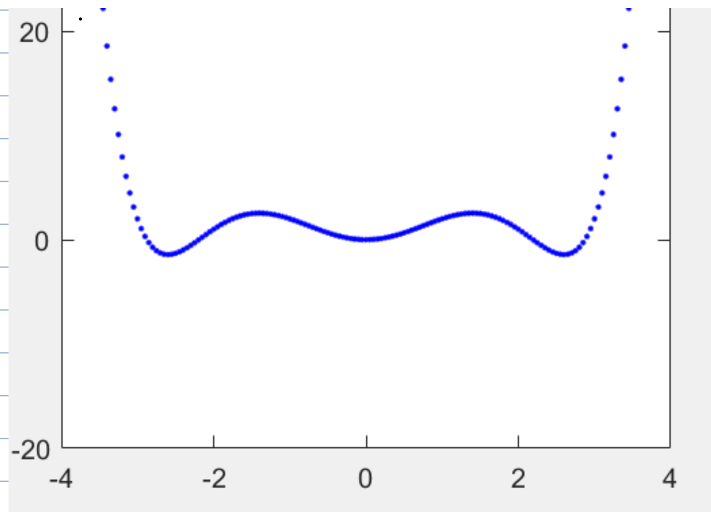
③ compute $\nabla F(x)^T F(x)$ ..... $O(d^2)$ (d^2 times multipication)

So per iteration cost should be $O(d^2)$, faster than Newton's method

Newton's method : $O(d^3)$ per iteration

example:

$$f(x) = \frac{517}{180}x^2 - \frac{17}{18}x^4 + \frac{13}{180}x^6$$



it passes (0,0) (1,2) (2,1) (3,2)
and symmetric along y-axis

$$\nabla f(x) = \frac{517}{90}x - \frac{34}{9}x^3 + \frac{13}{30}x^5$$

$$\nabla^2 f(x) = \frac{517}{90} - \frac{34}{3}x^2 + \frac{13}{6}x^4, = F(x)$$

it may stop at (0,0) if x_0 is near (0,0) and $F(x^*) \neq 0$ and $F(x) = 0$ exist

x^*

c) $\frac{1}{2}F(y)^T F(y) \geq \frac{1}{2}F(x)^T F(x) + \nabla F(x) \cdot F(x)^T (y-x)$ for $\forall, x,y \in R^n$

hessian of f

$\Rightarrow f$ is a convex

3) a) let $F(x,\lambda) = \begin{bmatrix} (A-\lambda I)x \\ x^T x - 1 \end{bmatrix}$

then, $\nabla F(x,\lambda) = \begin{pmatrix} (A-\lambda I) & -x \\ 2x^T & 0 \end{pmatrix}$

Similar to newton's method: F isn't "F" func above.

$$\left.\begin{array}{l} \text{Pick } X_{k+1} \text{ s.t } F(X_k) + \nabla F(k+1) \cdot (X_{k+1} - X_k) = 0 \\ \text{So } X_{k+1} = X_k - \nabla F(X_k)^{-1} F(X_k) \end{array}\right\} \text{(Newton's method)}$$

for this problem we can do: $\begin{pmatrix} X_{k+1} \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} X_k \\ \lambda_k \end{pmatrix} - \nabla F(X_k,\lambda_k)^{-1} F(X_k,\lambda_k)$

y is the combination of x and lambda

In [11]:

```
iter_max = 100
y_list = []
y_now =[]
y_next =np.array( [1/5 ,-1/5 ,4/5,1]).reshape(-1,1)
# y is the combination of x and lambda

Lambda = lambda0
x_now = x0
x_next = []

steps = 0

while np.abs( Lambda - eigvalues[1]) > 1e-3 :

    y_now = y_next
    y_next = y_now - grad_F_inv(x_now,Lambda)@F(x_now,Lambda)
    x_now = y_next[0:3 , :]
    Lambda = y_next[3]

    y_list.append(y_next)
    steps = steps+1

print("last y: \n",y_list[-1])
print("iteration steps: ", steps)
```

```
last y:
 [[ 0.8652295 ]
 [-1.46862935]
 [ 1.05211505]
 [ 1.82128088]]
iteration steps:  8
```

b)

```
Lambda = lambda0
x_now = x0
x_next = []

steps = 0

eta = 0.1
tau = 0.9
alpha = 100

while np.abs( Lambda - eigvalues[1]) > 1e-3 :

    y_now = y_next

    alpha_now = alpha # initial step size

    while True:
        a = h(y_now - alpha_now*grad_h(x_now,Lambda) )
        b = (h(y_now)- eta*alpha_now*(np.linalg.norm(grad_h(x_now,Lambda))**2 ))
        if(a<b):
            break
        alpha_now = alpha_now*tau

    y_next = y_now - alpha_now*grad_h(x_now,Lambda)

    x_now = y_next[0:3 , :]
    Lambda = y_next[3]

    y_list.append(y_next)
    steps = steps+1

print("last y: \n",y_list[-1])
print("iteration times: ", steps)
```

```
last y:
 [[ 0.43199866]
 [-0.73294044]
 [ 0.5255232 ]
 [ 1.82088483]]
iteration times:  106
```

compare to Newton's method, the algo proposed in 2 b) converges slower, and thus takes more iteration.