# Solutions to Homework 1

## 1 Problem 1

### a

The model parameter is $m = [P, d]^T$;

### b

$[\frac{\partial u}{\partial P}, \frac{\partial u}{\partial d}] = [\frac{d}{(x^2+y^2+d^2)^{3/2}}, \frac{P}{(x^2+y^2+d^2)^{3/2}} - \frac{3Pd^2}{(x^2+y^2+d^2)^{5/2}}]$;

### c

$$\begin{bmatrix} \frac{\partial^2 u}{\partial P^2} & \frac{\partial^2 u}{\partial P \partial d} \\ \frac{\partial^2 u}{\partial d \partial P} & \frac{\partial^2 u}{\partial d^2} \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{(x^2+y^2+d^2)^{3/2}} - \frac{3d^2}{(x^2+y^2+d^2)^{5/2}} \\ \frac{1}{(x^2+y^2+d^2)^{3/2}} - \frac{3d^2}{(x^2+y^2+d^2)^{5/2}} & \frac{15Pd^3}{(x^2+y^2+d^2)^{7/2}} - \frac{9Pd}{(x^2+y^2+d^2)^{5/2}} \end{bmatrix}$$

Note that the $\frac{\partial^2 u}{\partial P \partial d} = \frac{\partial^2 u}{\partial d \partial P}$ given that the function $u(P, d)$ is smooth enough.

### d

No since the two chambers may have different locations$(x, y, d)$ and different P.

The model parameters for two magma chambers should be $[P_1, P_2, d_1, d_2, x_1, x_2, y_1, y_2]^T$ and the observed vertical displacement should be:

$u = \frac{P_1 d_1}{((x-x_1)^2+(y-y_1)^2+d_1^2)^{3/2}} + \frac{P_2 d_2}{((x-x_2)^2+(y-y_2)^2+d_2^2)^{3/2}}$

Since if there are two magma chambers , it is not easy to tell their locations $(x, y)$ in the first place.

So they need to be inverted from observations. Also we need to have 2 different depth (d) parameters and P(pressure) parameters to describe the two magma chamber.

### e

8

### f

No. Because there are 8 degrees of freedom in the model space while only 6 degrees of freedom in the data space.

To map from a 8 dimensional model space to 6 dimensional data space means that there will be different models being mapped to the same point in data space.

This causes the inversion to have multiple solutions and trade off between different parameters.

## 2 Problem 2

### a

Let the matrix $A = [\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4]$, then

$$\begin{aligned} A &= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \\ A^T A &= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 2 & 0 & 2 \\ 0 & 0 & 2 & 0 \\ 1 & 2 & 0 & 3 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{x}_1^T\boldsymbol{x}_1 & \boldsymbol{x}_1^T\boldsymbol{x}_2 & \boldsymbol{x}_1^T\boldsymbol{x}_3 & \boldsymbol{x}_1^T\boldsymbol{x}_4 \\ & \boldsymbol{x}_2^T\boldsymbol{x}_2 & \boldsymbol{x}_2^T\boldsymbol{x}_3 & \boldsymbol{x}_2^T\boldsymbol{x}_4 \\ & \text{sym.} & \boldsymbol{x}_3^T\boldsymbol{x}_3 & \boldsymbol{x}_3^T\boldsymbol{x}_4 \\ & & & \boldsymbol{x}_4^T\boldsymbol{x}_4 \end{bmatrix} \end{aligned}$$

Therefore, we see that $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$ are orthogonal to each other; and $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$ are orthogonal.

### b

There are two independent vectors in $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_4\}$, choosing any two from them, along with $\boldsymbol{x}_3$, which is independent with all of them, form a basis. All the bases are:

$$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3\},\ \{\boldsymbol{x}_1, \boldsymbol{x}_4, \boldsymbol{x}_3\},\ \{\boldsymbol{x}_2, \boldsymbol{x}_4, \boldsymbol{x}_3\}$$

### c

$$A \begin{bmatrix} \dfrac{1}{\lambda_1}\boldsymbol{x}_1 & \dfrac{1}{\lambda_2}\boldsymbol{x}_2 & \dfrac{1}{\lambda_3}\boldsymbol{x}_3 \end{bmatrix} = [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \boldsymbol{x}_3]$$

Let

$$\begin{aligned} B &= \begin{bmatrix} \frac{1}{\lambda_1}\boldsymbol{x}_1 & \frac{1}{\lambda_2}\boldsymbol{x}_2 & \frac{1}{\lambda_3}\boldsymbol{x}_3 \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{5} & -\frac{1}{4} \\ 0 & \frac{1}{5} & \frac{1}{4} \end{bmatrix} \\ C &= [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \boldsymbol{x}_3] \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix} \end{aligned}$$

Then

$$A = CB^{-1}$$

Since

$$det(B) = \quad 1/20$$

$$\begin{aligned} B^{-1} = & \; 20 \begin{bmatrix} \frac{1}{10} & 0 & 0 \\ 0 & \frac{1}{8} & -\frac{1}{10} \\ 0 & \frac{1}{8} & \frac{1}{10} \end{bmatrix}^T \\ = & \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2.5 & 2.5 \\ 0 & -2 & 2 \end{bmatrix} \end{aligned}$$

Then

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2.5 & 2.5 \\ 0 & -2 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4.5 & 0.5 \\ 0 & 0.5 & 4.5 \end{bmatrix}$$

## d

Since A is an symmetric matrix, we have $A = Q\Lambda Q^{-1}$, where

$$Q = [\tilde{\boldsymbol{x}}_1 \quad \tilde{\boldsymbol{x}}_2 \quad \tilde{\boldsymbol{x}}_3]$$

$\tilde{\boldsymbol{x}}_i$ is the normalized $\boldsymbol{x}_i$.

In this problem,

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ 0 & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

It's a 45-degree axial rotation along $\boldsymbol{e}_1$.

$$\Lambda = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

It's stretching along three standard basis directions.
Therefore, the transformation includes: (1) A -45 degree axial rotation along $\boldsymbol{e}_1$; (2) 2, 5 and 4 times stretching along the three standard basis directions; (3) A 45 degree axial rotation back.
Check it with $\boldsymbol{x} = [1 \quad 2 \quad 3]^T$.

$$\begin{aligned} v_1 = & \; Q^{-1}x = [1 \quad 3.5355 \quad 0.7071]^T \\ v_2 = & \; \Lambda v_1 = [2 \quad 17.6777 \quad 2.8284]^T \\ v_3 = & \; Qv_2 = [2 \quad 10.5 \quad 14.5] = Ax \end{aligned}$$

# 3 Problem 3

### a

This implies $v^T x = 0$ since $v \in V$ means that $v$ is orthogonal to $x$.

### b

This is simply showing that the set $V$ is a subspace of $R^n$

$(a_1 v_1 + a_2 v_2)^T x = a_1 v_1^T x + a_2 v_2^T x = a_1(0) + a_2(0) = 0$

This imply that $(a_1 v_1 + a_2 v_2)$ is also orthogonal to $x$.

so $(a_1 v_1 + a_2 v_2) \in V$

### c

Since $x$ is an eigen vector of $A$, we assume $Ax = kx$ where $k \in R$

$(Av)^T x = v^T A^T x = v^T Ax = v^T(Ax) = v^T kx = kv^T x = 0$

So $Av$ is also orthogonal to $x$ $\forall v \in V$

### d

Given condition: $\forall v \in V (V \neq \{0\})$,if $Av \in V$ then one can find one eigen vector of $A$ in subspace $V$.

$A$ is a $n \times n$ matrix.

1) one can find one eigen vector $x$ in $R^n$. Since $R^n$satisfy Given condition. This is trivial.

2) if a subspace $V$ satisfy Given condition, and we have an eigen vector $x \in V$. Then we prove that the set $U = \{u | u^T x = 0, u \in V\}$ also satisfy Given condition.

It is shown in (b) that $U$ forms subspace .

$\forall u \in U$,$Au \in V$ since $U \subseteq V$

we can write $Au = cx + w, w \in U, c \in R$, multiply both sides by $x^T$:

$x^T Au = cx^T x + x^T w = cx^T x$

note that on the left hand side:$x^T Au = u^T A^T x = u^T Ax = ku^T x = 0$

So $cx^T x = c|x|^2 = 0$,which implies that $c = 0$,

So $Au = w \in U, \forall u \in U$. So $U$ satisfy Given condition.

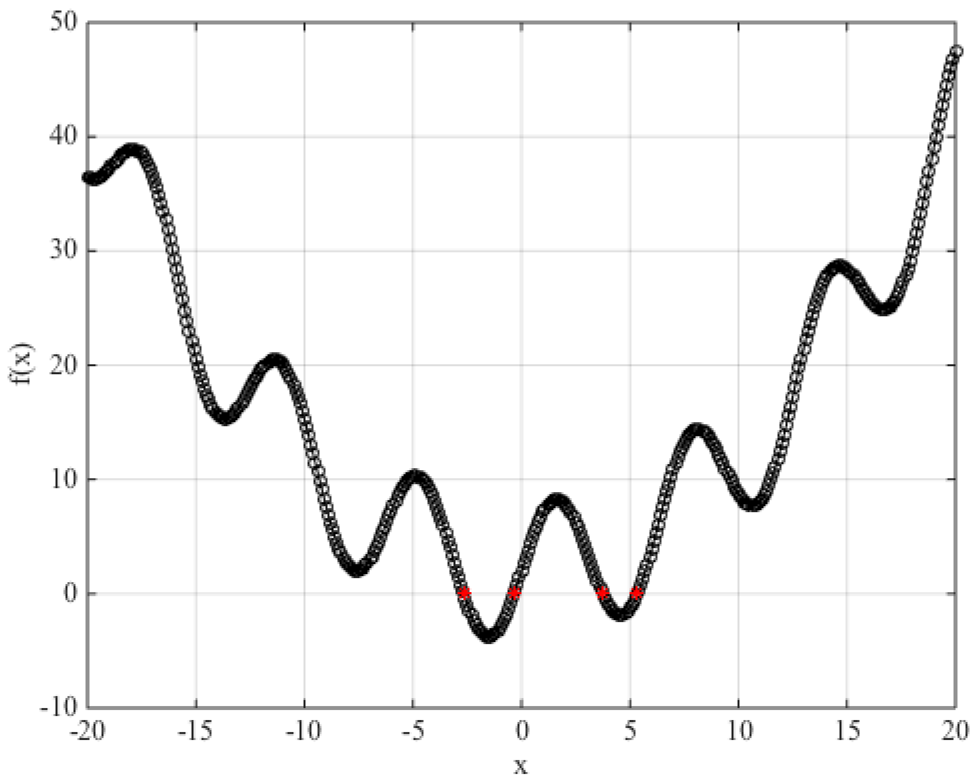
3) That means we can always find one eigen vector while the remaining subspace still satisfy Given condition until it shrinks to $\{0\}$. Since each time , the dimension of the space is decreased by 1, we can find n orthogonal eigen vectors.

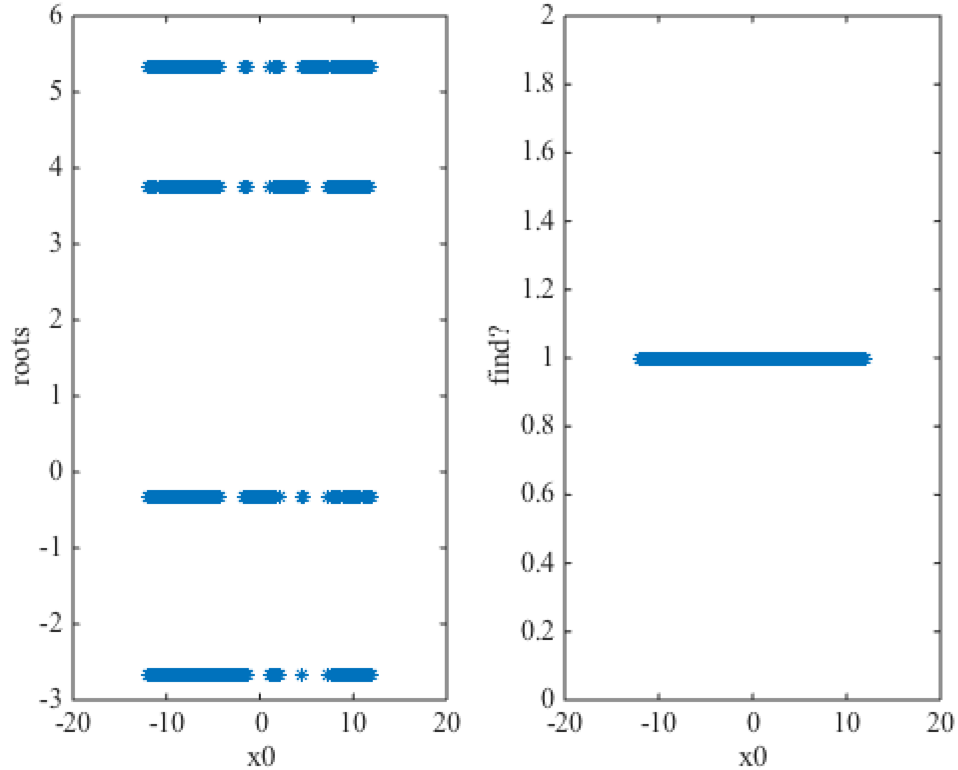
# 4 Problem 4

There is no evidence that Newton's method will find the root closest to the initial guess. Because the "jump" in x is largely controlled by the gradient at current point, a small gradient will cause a large jump to the neighbor of another root.

```
function hw1_4
set(0,'defaulttextfontname','times','defaulttextfontsize',14);
set(0,'defaultaxesfontname','times','defaultaxesfontsize',14);

x = -20:0.1:20;
f = f_func(x);

figure(1)
plot(x,f,'ko-');
grid on; xlabel('x'); ylabel('f(x)');

epsilon = 1e-10;
x0 = -12:0.01:12;
itermax = 1000;

roots = zeros(length(x0),1);
flags = zeros(length(x0),1);
for i = 1:length(x0)
    [roots(i), flags(i)] = find_root(x0(i),epsilon,itermax);
end


figure(1)
hold on;
plot(roots,f_func(roots),'r*');
hold off;


figure(2)

subplot(121);
plot(x0,roots,'*'); xlabel('x0'); ylabel('roots');
subplot(122)
plot(x0,flags,'*'); xlabel('x0'); ylabel('find?');

end


function [root,flag] = find_root(x0,epsilon,itermax)
root = x0;
flag = false;   % whether you find a root
for i = 1:itermax
    f_curr = f_func(x0);
    f_grad_curr = f_grad_func(x0);

    if abs(f_grad_curr) < epsilon   % denominator is too small, stop
       break;
    end

    x1 = x0 - f_curr/f_grad_curr;

    if abs( x1-x0 ) < epsilon
```

```
        flag = true;      % find the root
        root = x0;
        break;
    end

    x0 = x1;
end
end

% value of the function at x
function val = f_func(x)
val = x.^2/10 + 6 * sin(x) + 2;
end

% value of the gradient at x
function val = f_grad_func(x)
val = x./5 + 6 * cos(x);
end
```






*Published with MATLAB® R2014b*