HW3 — solution

1. A closed form approach to ellipse parameter determination is:

$$\begin{vmatrix} x^2 & y^2 & x & y & 1 \\ x_1^2 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & y_3^2 & x_3 & y_3 & 1 \\ x_4^2 & y_4^2 & x_4 & y_4 & 1 \end{vmatrix} = 0$$

Evaluate this determinant for the 4 given points yields $Ax^2 + By^2 + Cx + Dy + E = 0$ *

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = \begin{bmatrix} -.3582 \\ -.9352 \\ .1589 \\ .8505 \\ 5.0120 \end{bmatrix}$$ mult. by -1 so coeff's of A & B are positive, $$\begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = \begin{bmatrix} .3582 \\ .9352 \\ -.1589 \\ -.8505 \\ -5.0120 \end{bmatrix}$$

expand $\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1 \Rightarrow \frac{x^2}{a^2} - \frac{2xx_0}{a^2} + \frac{x_0^2}{a^2} + \frac{y^2}{b^2} - \frac{2yy_0}{b^2} + \frac{y_0^2}{b^2} = 1$

The $A \to E$ are known only to a scale factor, k. Equate like terms between 2 expressions:

$$Ak = \frac{1}{a^2}, \quad Ck = -\frac{2x_0}{a^2}, \quad Ek = -1 + \frac{x_0^2}{a^2} + \frac{y_0^2}{b^2}$$

$$Bk = \frac{1}{b^2}, \quad Dk = -\frac{2y_0}{b^2},$$ rearrange, solve for x_0, y_0

$$C\cancel{k} = -2x_0 \cdot A\cancel{k}, \quad C = -2x_0 A, \quad x_0 = -C/2A = \underline{0.2217}$$

$$D\cancel{k} = -2y_0 \cdot B\cancel{k}, \quad D = -2y_0 B, \quad y_0 = -D/2B = \underline{0.4547}$$

$$Ek = -1 + Akx_0^2 + Bky_0^2, \quad 1 = Akx_0^2 + Bky_0^2 - Ek$$

$$1 = k(Ax_0^2 + By_0^2 - E)$$

$$k = 1/(Ax_0^2 + By_0^2 - E) = \underline{0.1915}$$

$$a = \sqrt{1/AK} = \underline{3.8185}$$ (results same as newton iteration)

$$b = \sqrt{1/BK} = \underline{2.3633}$$

---

* $M = \begin{bmatrix} x_1^2 & y_1^2 & x_1 & y_1 & 1 \\ x_2^2 & y_2^2 & x_2 & y_2 & 1 \\ x_3^2 & y_3^2 & x_3 & y_3 & 1 \\ x_4^2 & y_4^2 & x_4 & y_4 & 1 \end{bmatrix}$,

```
M1 = M;
M1(:,1) = [];
A = det(M1);
```

etc. for other coefficients

to solve by newton iteration (4 equations in 4 unknowns)

$$F_i = \frac{(x_i - x_0)^2}{a^2} + \frac{(y_i - y_0)^2}{b^2} - 1 = 0 \quad , \quad i: 1 \to 4$$

$$\frac{\partial F}{\partial x_0} = -\frac{2(x-x_0)}{a^2} \quad , \quad \frac{\partial F}{\partial a} = -2(x-x_0)^2 a^{-3} \quad ,$$

$$\frac{\partial F}{\partial y_0} = -\frac{2(y-y_0)}{b^2} \quad , \quad \frac{\partial F}{\partial b} = -2(y-y_0)^2 b^{-3} \quad ,$$

evaluate @ current approximations of x_0, y_0, a, b

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} + \begin{bmatrix} \partial F_1/\partial a & \partial F_1/\partial b & \partial F_1/\partial x_0 & \partial F_1/\partial y_0 \\ \partial F_2/\partial a & \partial F_2/\partial b & \partial F_2/\partial x_0 & \partial F_2/\partial y_0 \\ \partial F_3/\partial a & \partial F_3/\partial b & \partial F_3/\partial x_0 & \partial F_3/\partial y_0 \\ \partial F_4/\partial a & \partial F_4/\partial b & \partial F_4/\partial x_0 & \partial F_4/\partial y_0 \end{bmatrix} \begin{bmatrix} \Delta a \\ \Delta b \\ \Delta x_0 \\ \Delta y_0 \end{bmatrix} = 0$$

$$F + J\Delta = 0$$

solve each iteration by,

$$\Delta = -J^{-1} F$$

then update

$$\begin{bmatrix} a^0 \\ b^0 \\ x_0^0 \\ y_0^0 \end{bmatrix}_{i+1} = \begin{bmatrix} a^0 \\ b^0 \\ x_0^0 \\ y_0^0 \end{bmatrix}_i + \begin{bmatrix} \Delta a \\ \Delta b \\ \Delta x_0 \\ \Delta y_0 \end{bmatrix}$$

& repeat until Δ vector is small

See matlab listing

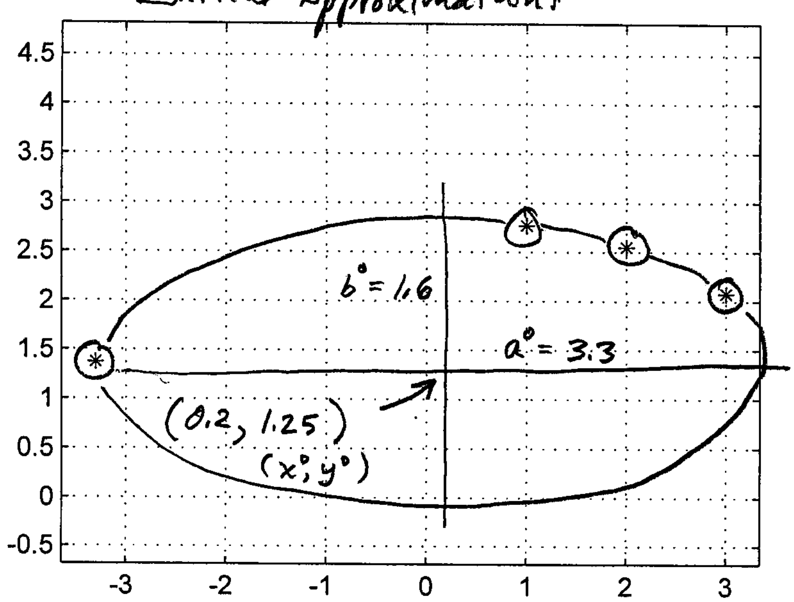
```
% hw3_1_sol.m  23-sep-2011
% find ellipse parameters using newton iteration

a0=3.3;
b0=1.6;
x0=0.2;
y0=1.25;

x=[1.0;2.0;3.0;-3.3];
y=[2.7684;2.5461;2.076;1.3682];

J=zeros(4,4);
F=zeros(4,1);

for iter=1:8
  for i=1:4
    dF_da=-2*(x(i)-x0)^2 * a0^-3;
    dF_db=-2*(y(i)-y0)^2 * b0^-3;
    dF_dx0=-2*(x(i)-x0)/a0^2;
    dF_dy0=-2*(y(i)-y0)/b0^2;
    J(i,:)=[dF_da dF_db dF_dx0 dF_dy0];
    F(i)=(x(i)-x0)^2/a0^2 + (y(i)-y0)^2/b0^2 -1;
    end
  iter
  %F
  del=-inv(J)*F;
  del'
  a0=a0 + del(1);
  b0=b0 + del(2);
  x0=x0 + del(3);
  y0=y0 + del(4);
  end

% check manually for convergence

a0
b0
x0
y0
```

hw3_1_sol

Output

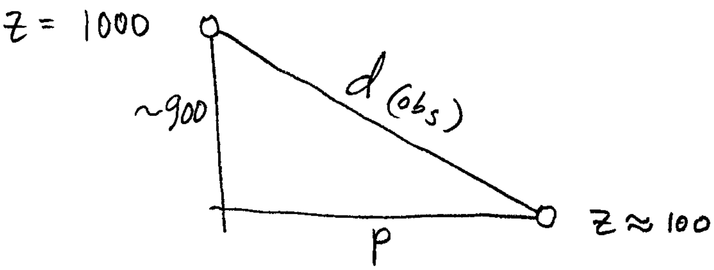
```
hw3_1_sol
iter =
     1
ans =
    0.2603    0.3865    0.0183   -0.4111
iter =
     2
ans =
    0.1906    0.2724    0.0031   -0.2788
iter =
     3
ans =
    0.0622    0.0955    0.0004   -0.0965
iter =
     4
ans =
    0.0054    0.0088    0.0000   -0.0088
iter =
     5
ans =
  1.0e-004 *
    0.3843    0.6646    0.0003   -0.6662
iter =
     6
ans =
  1.0e-008 *
    0.2098    0.3743    0.0001   -0.3747
iter =
     7
ans =
  1.0e-013 *
   -0.0638   -0.1247   -0.0121    0.1240
iter =
     8
ans =
  1.0e-013 *
    0.1436    0.2654    0.0271   -0.2684
a0 =
    3.8185
b0 =
    2.3633
x0 =
    0.2217
y0 =
    0.4547
diary off
```

Initial Approximations

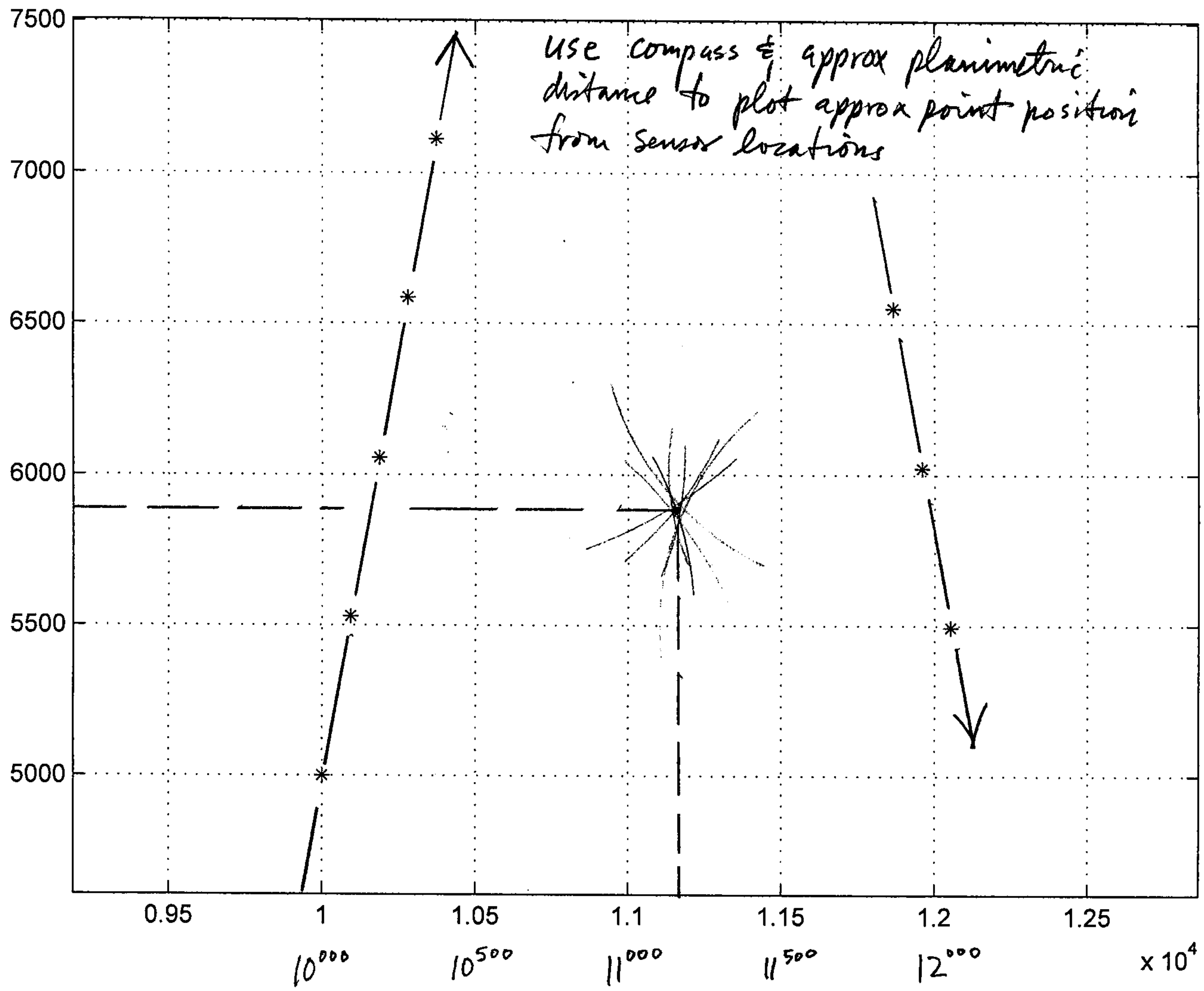


Initial Approximations

d	p
1721	1147
1448	1134
1345	1000
1447	1134
1720	1466
1318	962
1204	800
1317	962



plot arcs @ planimetric distances



approx. location : 11,200 (X)

5,900 (Y)

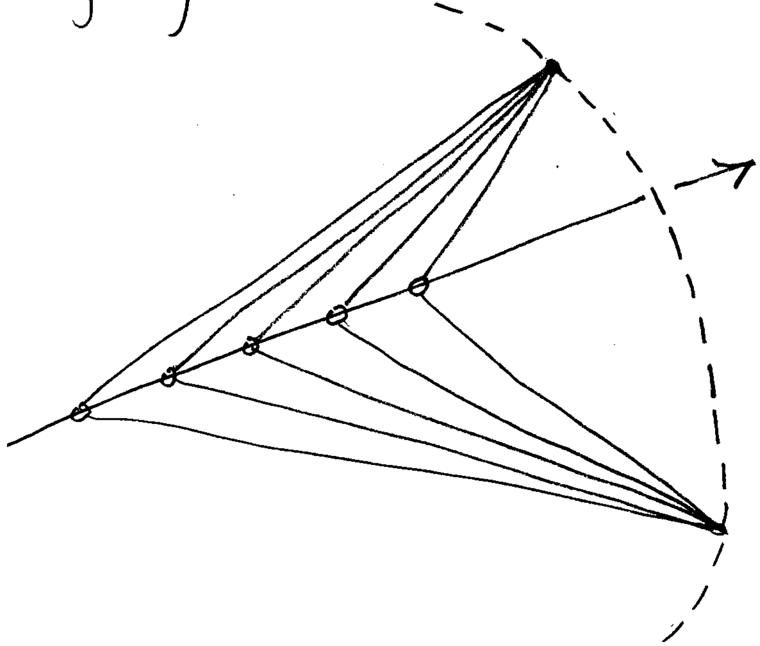
100 (Z)

```
% hw3_2_sol.m  23-sep-2011
% solve 3D ranging problem

n=8;
n0=3;
r=5;
u=n0;
c=n;
W=eye(8);

% indirect observations

d=[1721.09;1448.17;1345.76;1447.97;1720.89;1318.05;1204.46;1317.85];

% initial approx
x0=11200;
y0=5900;
z0=100;

X=zeros(8,1);
Y=zeros(8,1);
Z=zeros(8,1);
X(1)=10000.0;
Y(1)=5000.0;
Z(1)=1000.0;
V1=[9.315322;52.829815;0];
for i=1:4
  ii=i+1;
  X(ii)=X(1) + i*10*V1(1);
  Y(ii)=Y(1) + i*10*V1(2);
  Z(ii)=Z(i) + i*10*V1(3);
  end
X(6)=11865.808;
Y(6)=6550.165;
Z(6)=1000;
V2=[9.315322;-52.829815;0];
for i=1:2
  ii=i+6;
  X(ii)=X(6) + i*10*V2(1);
  Y(ii)=Y(6) + i*10*V2(2);
  Z(ii)=Z(6) + i*10*V2(3);
  end

n_iter=1;
keep_going=1;
while((keep_going == 1) & (n_iter <= 10))
  B=zeros(c,u);
  f=zeros(c,1);
  for i=1:c
    Di=sqrt((x0-X(i))^2 + (y0-Y(i))^2 + (z0-Z(i))^2);
    dFdx=-(x0-X(i))/Di;
    dFdy=-(y0-Y(i))/Di;
    dFdz=-(z0-Z(i))/Di;
    B(i,:)=[dFdx dFdy dFdz];
    F=d(i) - Di;
    f(i)=-F;
    end
  del=inv(B'*W*B)*B'*W*f
  x0=x0 + del(1);
  y0=y0 + del(2);
  z0=z0 + del(3);
  if(all(abs(del) < 0.00001))
    keep_going = 0;
    end
  n_iter=n_iter + 1;
  end

if(keep_going == 0)
  disp('solution converged');
  [x0; y0; z0]
else
  disp('solution did not converge');
  end
```

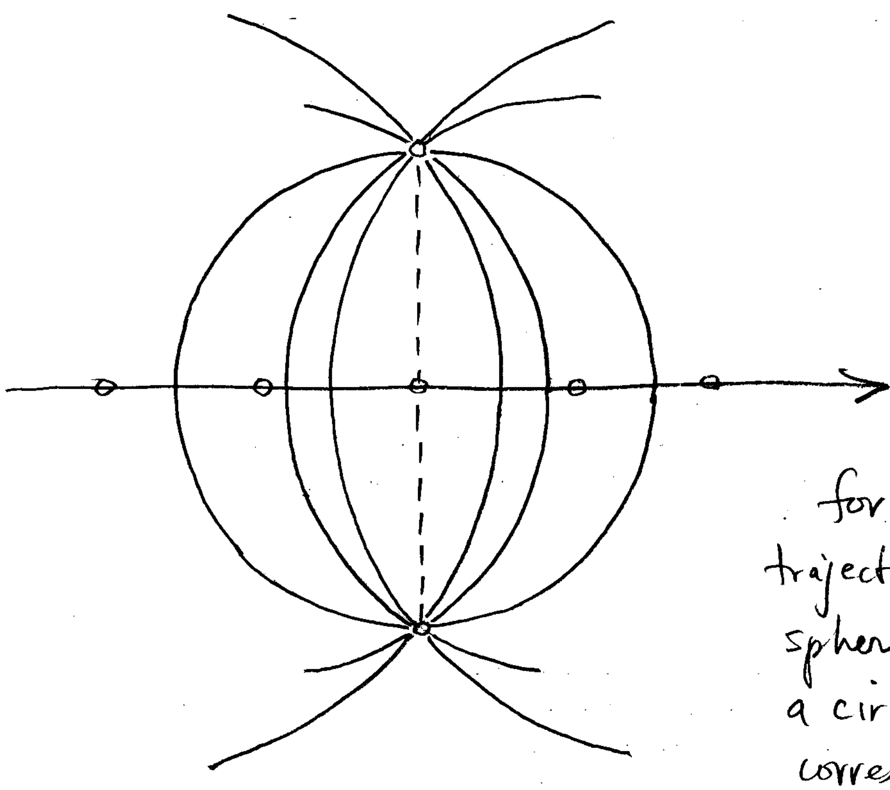
Output

```
hw3_2_sol
del =
          -28.791696184914
          -17.0915954991766
          -0.359217761854023
del =
          0.0111088551324292
          0.116763553391987
          0.389578360988621
del =
     -1.64433796771973e-005
     -1.83466179663333e-005
      4.70895153268924e-005
del =
      1.08205375215498e-010
      8.51690384884307e-011
     -9.60455629782715e-011
solution converged
ans =
          11171.2193962269
          5883.02514970768
          100.030407688554

v =
   -0.2084
    0.2785
   -0.3509
    0.4088
   -0.1258
    0.0828
   -0.4004
    0.3297
```

Why single trajectory fails?



The "fan" of ranges can rotate about the sensor trajectory ⇒ rank of B & N becomes 2



for point "beneath" trajectory the 5 range spheres intersect along a circle (¦) which corresponds to the circle in the first sketch