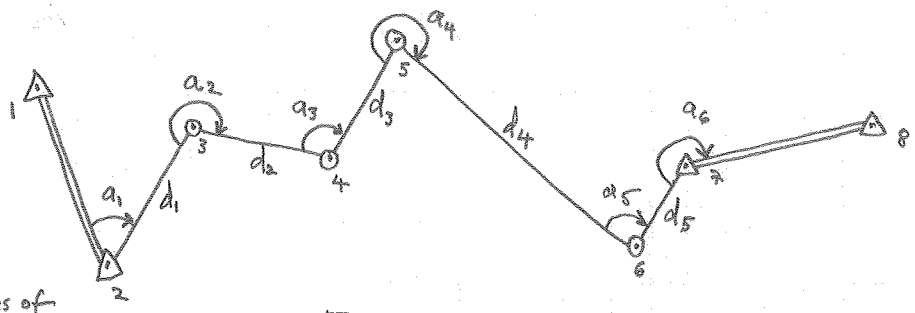
HW3 Solution

$n = 11$

$n_0 = 8$

$r = 3$



Indirect Obs. $\Rightarrow$ $m = n_0 = 8$, use $x_3, y_3, x_4, y_4, x_5, y_5, x_6, y_6$

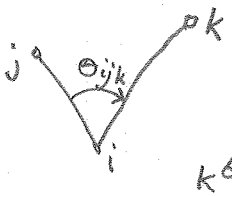
types of condition equations

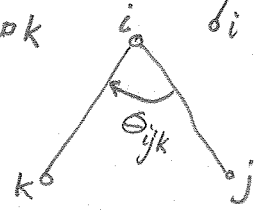
$$F_d = d_{ij} - \left[(x_j - x_i)^2 + (y_j - y_i)^2\right]^{1/2}$$

$$\partial F_d/\partial x_i = \Delta x/D_{ij}\,, \quad \partial F_d/\partial y_i = \Delta y/D_{ij}\,, \quad \partial F_d/\partial x_j = -\Delta x/D_{ij}\,, \quad \partial F_d/\partial y_j = -\Delta y/D_{ij}$$

$\Delta x = x_j - x_i$, $\Delta y = y_j - y_i$; $f = -F_d$

$$F_\theta = \theta_{ijk} - \left[\text{atan}\,\frac{\Delta x_{ik}}{\Delta y_{ik}} - \text{atan}\,\frac{\Delta x_{ij}}{\Delta y_{ij}}\right]$$



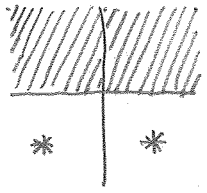


$$\partial F_\theta/\partial x_i = \frac{-\Delta y_{ij}}{D_{ij}^2} + \frac{\Delta y_{ik}}{D_{ik}^2}\,, \quad \partial F_\theta/\partial y_i = \frac{\Delta x_{ij}}{D_{ij}^2} - \frac{\Delta x_{ik}}{D_{ik}^2}\,,$$

$$\partial F_\theta/\partial x_j = \frac{\Delta y_{ij}}{D_{ij}^2}\,, \quad \partial F_\theta/\partial y_j = \frac{-\Delta x_{ij}}{D_{ij}^2}\,, \quad \partial F_\theta/\partial x_k = \frac{-\Delta y_{ik}}{D_{ik}^2}\,, \quad \partial F_\theta/\partial y_k = \frac{\Delta x_{ik}}{D_{ik}^2}$$

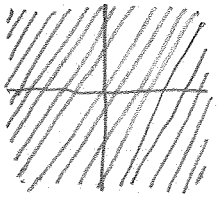
$\Delta x_{ij} = x_j - x_i$, $\Delta y_{ij} = y_j - y_i$, $\Delta x_{ik} = x_k - x_i$, $\Delta y_{ik} = y_k - y_i$, $f = -F_\theta$

→ use: $\left[\text{atan2}(\Delta x_{ik}, \Delta y_{ik}) - \text{atan2}(\Delta x_{ij}, \Delta y_{ij})\right]$ if [·] negative add 360° or 2π



$\text{atan}\left(\frac{\Delta x}{\Delta y}\right)$ returns angles only in shaded quadrants, you must fix * (add 180°, π)

atan2(Δx, Δy) does this for you and returns angles in all 4 quadrants



take a_4 as example = i is point 5, j is point 4, k is point 6

columns of B

x_3	y_3	x_4	y_4	x_5	y_5	x_6	y_6
0	0	$\frac{\Delta y_{54}}{D_{54}^2}$	$\frac{-\Delta x_{54}}{D_{54}^2}$	$\frac{-\Delta y_{54}}{D_{54}^2} + \frac{\Delta y_{56}}{D_{56}^2}$	$\frac{\Delta x_{54}}{D_{54}^2} - \frac{\Delta x_{56}}{D_{56}^2}$	$\frac{-\Delta y_{56}}{D_{56}^2}$	$\frac{\Delta x_{56}}{D_{56}^2}$

Similar strategy for all condition equations

```
hw3_sol
dist_sig =
    0.0090
    0.0088
    0.0088
    0.0096
    0.0085
W =
   1.0e+04 *
  Columns 1 through 9
    3.4673         0         0         0         0         0         0         0         0
         0    0.0001         0         0         0         0         0         0         0
         0         0    3.4673         0         0         0         0         0         0
         0         0         0    0.0001         0         0         0         0         0
         0         0         0         0    3.4673         0         0         0         0
         0         0         0         0         0    0.0001         0         0         0
         0         0         0         0         0         0    3.4673         0         0
         0         0         0         0         0         0         0    0.0001         0
         0         0         0         0         0         0         0         0    3.4673
         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0
  Columns 10 through 11
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
         0         0
    0.0001         0
         0    3.4673
n =
    11
n0 =
     8
r =
     3
u =
     8
c =
    11
ok, solve the LS problem
B =
   -0.0024    0.0016         0         0         0         0         0         0
   -0.5534   -0.8329         0         0         0         0         0         0
    0.0013   -0.0053    0.0011    0.0036         0         0         0         0
    0.9560   -0.2933   -0.9560    0.2933         0         0         0         0
    0.0011    0.0036    0.0022   -0.0053   -0.0033    0.0016         0         0
         0         0    0.4397    0.8981   -0.4397   -0.8981         0         0
         0         0   -0.0033    0.0016    0.0021   -0.0031    0.0013    0.0014
         0         0         0         0    0.7474   -0.6644   -0.7474    0.6644
         0         0         0         0    0.0013    0.0014    0.0036   -0.0043
         0         0         0         0         0         0    0.5004    0.8658
         0         0         0         0         0         0   -0.0049    0.0028
f =
   -0.0016
    0.8239
    0.0078
    1.5522
   -0.0145
    0.0085
    0.0043
    0.8136
    0.0020
   -1.3591
    0.0020
del =
    0.0108
   -0.9966
   -0.9952
    1.0166
    1.0067
    0.0267
   -0.9912
   -0.9966
iter =
     1
del =
    0.0011
    0.0008
   -0.0043
   -0.0062
   -0.0093
   -0.0124
   -0.0020
   -0.0001
iter =
     2
del =
   1.0e-06 *
    0.1006
   -0.0205
    0.0400
    0.1205
    0.1230
    0.0926
   -0.0691
   -0.0024
iter =
     3
del =
   1.0e-11 *
```

```
    0.0361
    0.0337
    0.0131
   -0.1812
    0.0517
   -0.0896
   -0.0041
   -0.0437
iter =
     4
we have converged
points
ans =
   1.0e+03 *
    1.5200    5.4850
ans =
   1.0e+03 *
    1.7700    5.4100
ans =
   1.0e+03 *
    1.8900    5.6500
ans =
   1.0e+03 *
    2.2750    5.3050
v =
    0.0000
    0.0015
   -0.0000
    0.0020
    0.0000
    0.0012
   -0.0000
    0.0016
    0.0000
    0.0013
    0.0000
va1_sec =
    2.9793
va2_sec =
   -0.4707
va3_sec =
    1.0104
va4_sec =
   -1.9714
va5_sec =
    3.3803
va6_sec =
    1.4785
adjusted observations
adj_dobs =
  342.5375
  260.9950
  268.3302
  516.9766
  179.2283
adj_aobs =
    0.8330
    4.4158
    1.7430
    4.9794
    1.3662
    3.9381
ans =
   47.0000   43.0000   39.2793
ans =
  253.0000         0   26.4293
ans =
   99.0000   51.0000   55.5104
ans =
  285.0000   17.0000   59.1286
ans =
   78.0000   16.0000   34.1803
ans =
  225.0000   38.0000    2.4785
diary off
```

```
% hw3_sol.m  13-oct-2014

% sub other approximations to compare 1st iteration results
x=[1240;1330;1520;1771;1889;2276;2365;2720];
y=[5560;5200;5486;5409;5650;5306;5460;5550];

% function [b,F,comp_obs]=ang(x,y,at,from,to,deg,min,sec)
% function [b,F,comp_obs]=dist2d(x,y,at,to,obs)
degrad=180/pi;

angdms=[ 47 43 36.3;
        253 00 26.9;
         99 51 54.5;
        285 18 01.1;
         78 16 30.8;
        225 38 01.0];

dobs=[342.536;260.993;268.329;516.975;179.227];

deg=0;
min=0;
sec=0;
obs=0;
ang_sigd=10/3600;
ang_sigr=ang_sigd/degrad;

dist_sig=[0.008+3e-06*dobs(1); 0.008+3e-06*dobs(2); 0.008+3e-06*dobs(3); 0.008+3e-06*dobs(4);
          0.008+3e-06*dobs(5)]
sig0=dist_sig(1);
wa=sig0^2/ang_sigr^2;
% each distance sigma is (slightly) different
wd=[sig0^2/dist_sig(1)^2; sig0^2/dist_sig(2)^2; sig0^2/dist_sig(3)^2; sig0^2/dist_sig(4)^2;
    sig0^2/dist_sig(5)^2];
W=diag([wa wd(1) wa wd(2) wa wd(3) wa wd(4) wa wd(5) wa])

n=11
n0=8
r=n-n0
u=n0
c=n

disp('ok, solve the LS problem');
x0=x;
y0=y;
keep_going=1;
iter=0;
thresh=1.0e-08;
while(keep_going == 1)
  iter=iter + 1;
  B=zeros(c,u);
  f=zeros(c,1);
  adms=angdms(1,:);
  [b,F,comp_obs]=ang(x0,y0,2,1,3,adms(1),adms(2),adms(3));
  B(1,:)=[b(5) b(6) 0 0 0 0 0 0];
  f(1)=-F;
  [b,F,comp_obs]=dist2d(x0,y0,2,3,dobs(1));
  B(2,:)=[b(3) b(4) 0 0 0 0 0 0];
  f(2)=-F;
  adms=angdms(2,:);
  [b,F,comp_obs]=ang(x0,y0,3,2,4,adms(1),adms(2),adms(3));
  B(3,:)=[b(1) b(2) b(5) b(6) 0 0 0 0];
  f(3)=-F;
  [b,F,comp_obs]=dist2d(x0,y0,3,4,dobs(2));
  B(4,:)=[b(1) b(2) b(3) b(4) 0 0 0 0];
  f(4)=-F;
  adms=angdms(3,:);
  [b,F,comp_obs]=ang(x0,y0,4,3,5,adms(1),adms(2),adms(3));
  B(5,:)=[b(3) b(4) b(1) b(2) b(5) b(6) 0 0];
  f(5)=-F;
  [b,F,comp_obs]=dist2d(x0,y0,4,5,dobs(3));
  B(6,:)=[0 0 b(1) b(2) b(3) b(4) 0 0];
  f(6)=-F;
  adms=angdms(4,:);
  [b,F,comp_obs]=ang(x0,y0,5,4,6,adms(1),adms(2),adms(3));
  B(7,:)=[0 0 b(3) b(4) b(1) b(2) b(5) b(6)];
  f(7)=-F;
  [b,F,comp_obs]=dist2d(x0,y0,5,6,dobs(4));
  B(8,:)=[0 0 0 0 b(1) b(2) b(3) b(4)];
  f(8)=-F;
  adms=angdms(5,:);
  [b,F,comp_obs]=ang(x0,y0,6,5,7,adms(1),adms(2),adms(3));
  B(9,:)=[0 0 0 0 b(3) b(4) b(1) b(2)];
  f(9)=-F;
  [b,F,comp_obs]=dist2d(x0,y0,6,7,dobs(5));
  B(10,:)=[0 0 0 0 0 0 b(1) b(2)];
  f(10)=-F;
  adms=angdms(6,:);
  [b,F,comp_obs]=ang(x0,y0,7,6,8,adms(1),adms(2),adms(3));
  B(11,:)=[0 0 0 0 0 0 b(3) b(4)];
  f(11)=-F;
  if(iter == 1)
    B
    f
    end

  N=B'*W*B;
  t=B'*W*f;
  f;
  del=inv(N)*t
  iter
  %pause
  x0(3)=x0(3)+del(1);
  y0(3)=y0(3)+del(2);
  x0(4)=x0(4)+del(3);
```

```
  y0(4)=y0(4)+del(4);
  x0(5)=x0(5)+del(5);
  y0(5)=y0(5)+del(6);
  x0(6)=x0(6)+del(7);
  y0(6)=y0(6)+del(8);
  v=f-B*del;
  if all(abs(del) < thresh)
    keep_going=0;
    disp('we have converged');
    end
  if(iter > 10)
    keep_going=0;
    disp('too many iterations');
    end
  end
disp('points');
[x0(3) y0(3)]
[x0(4) y0(4)]
[x0(5) y0(5)]
[x0(6) y0(6)]
v
va1_sec=v(1)*degrad*3600
va2_sec=v(3)*degrad*3600
va3_sec=v(5)*degrad*3600
va4_sec=v(7)*degrad*3600
va5_sec=v(9)*degrad*3600
va6_sec=v(11)*degrad*3600

disp('adjusted observations')

aobs=zeros(6,1);
for i=1:6
  aobs(i)=(angdms(i,1)+angdms(i,2)/60+angdms(i,3)/3600)/degrad;
  end

adj_aobs=[aobs(1)+v(1); aobs(2)+v(3); aobs(3)+v(5); aobs(4)+v(7); aobs(5)+v(9); aobs(6)+v(11)];
adj_dobs=[dobs(1)+v(2); dobs(2)+v(4); dobs(3)+v(6); dobs(4)+v(8); dobs(5)+v(10)];

adj_dobs
adj_aobs
for i=1:6
  adj_adms=r2dms(adj_aobs(i));
  adj_adms'
  end

test_stat=v'*W*v/sig0^2;
```

```
% dist2d.m.m  6-nov-02
% function to evaluate distance condition equation
% and return elements of B-matrix, F, and computed obs
% function [b,F,cmpobs]=dist2d(x,y,at,to,obs)
% F = obs - sqrt((xt-xa)^2 + (yt-ya)^2) = 0
% order of unkowns: xa,ya,xt,yt
% args
%  x : array of x-coords of network points
%  y : array of y-coords of network points
%  at : index of "at" point
%  to : index of "to" point
%  obs : the distance observation

function [b,F,comp_obs]=dist2d(x,y,at,to,obs)
b=zeros(1,4);
dobs=obs;
xa=x(at);
ya=y(at);
xt=x(to);
yt=y(to);

dx=xt-xa;
dy=yt-ya;
D0=sqrt(dx^2+dy^2);
b(1)=(xt-xa)/D0;
b(2)=(yt-ya)/D0;
b(3)=-(xt-xa)/D0;
b(4)=-(yt-ya)/D0;
F=dobs - D0;
comp_obs=D0;
```

```
% ang.m  25-oct-06
% function to evaluate angle condition equation
% and return elements of B-matrix, F, and computed obs
% function [b,F,comp_obs]=ang(x,y,z,at,to,degree,minute,second)
% F_ang = theta - (atan((xk-xi)/(yk-yi)) - atan((xj-xi)/(yj-yi)) = 0
% at = i, from = j, to = k
% order of unknowns: xi,yi,xj,yj,xk,yk or xat,yat,xfrom,yfrom,xto,yto
% args
%  x : array of x-coords of network points
%  y : array of y-coords of network points
%  at : index of "at"point
%  from : index of "from" point
%  to : index of "to" point
%  degree,minute,second: d,m,s of direction observation

function [b,F,comp_obs]=ang(x,y,at,from,to,degree,minute,second)
degrad=180/pi;
xi=x(at);
yi=y(at);
xj=x(from);
yj=y(from);
xk=x(to);
yk=y(to);

dx_ij=xj-xi;
dy_ij=yj-yi;
dx_ik=xk-xi;
dy_ik=yk-yi;
D2_ij=dx_ij^2 + dy_ij^2;
D2_ik=dx_ik^2 + dy_ik^2;

dF_dxi= dy_ik/D2_ik - dy_ij/D2_ij;
dF_dyi=-dx_ik/D2_ik + dx_ij/D2_ij;
dF_dxj= dy_ij/D2_ij;
dF_dyj=-dx_ij/D2_ij;
dF_dxk=-dy_ik/D2_ik;
dF_dyk= dx_ik/D2_ik;
b=[dF_dxi dF_dyi dF_dxj dF_dyj dF_dxk dF_dyk];

az_ij=atan2(dx_ij,dy_ij);
az_ik=atan2(dx_ik,dy_ik);
angle=az_ik-az_ij;
if(angle < 0.0)
  angle=angle + 2*pi;
  end
theta=angle;
comp_obs=theta;
aobs=(degree + minute/60.0 + second/3600.0)/degrad;
F=aobs - comp_obs;
```