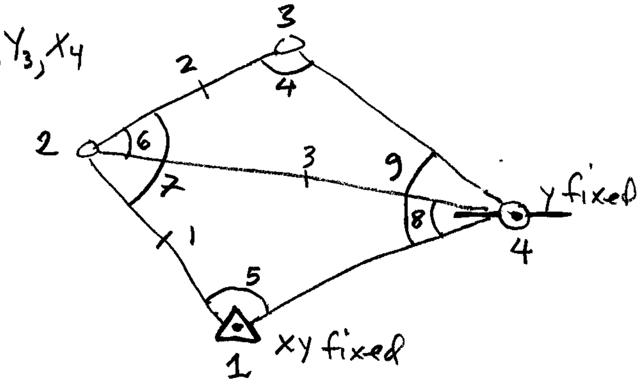
4-1

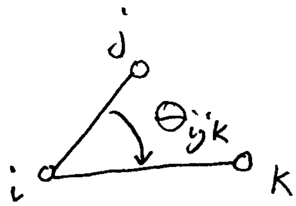
$n = 9$
$n_0 = 5$
$r = 4$

params = X_2, Y_2, X_3, Y_3, X_4

hw4_1_out




```
hw4_1_sol
del =
       -1.3054
      -0.61782
      -0.89097
       0.28932
       -1.0734
del =
    -0.0019254
     -0.004136
    -0.0041785
   -0.00075073
    -0.0019184
del =
  -8.7057e-008
  -8.4239e-008
  -1.9597e-007
   3.1165e-008
  -1.0328e-007
we have converged
v =
       -0.0107
      -0.13323
       0.10613
   -0.00010654
  -8.1134e-005
  -4.5578e-005
   -8.073e-005
  -9.3154e-005
   2.4062e-005
angle residuals in degrees
vd =
    -0.0061046
    -0.0046486
    -0.0026114
    -0.0046255
    -0.0053373
     0.0013787
lhat =
        447.48
        439.65
        597.05
        1.6929
        1.5402
       0.62906
        1.3837
       0.84676
        1.6664
adjusted angles in degrees
lha =
        96.997
        88.244
        36.042
        79.282
        48.516
        95.476
```



$$\frac{\partial F}{\partial x_i} = \frac{\Delta y_{ik}}{D^2_{ik}} - \frac{\Delta y_{ij}}{D^2_{ij}}$$

$$\frac{\partial F}{\partial y_i} = \frac{-\Delta x_{ik}}{D^2_{ik}} + \frac{\Delta x_{ij}}{D^2_{ij}}$$

$$\frac{\partial F}{\partial x_j} = \frac{\Delta y_{ij}}{D^2_{ij}}$$

$$\frac{\partial F}{\partial y_j} = \frac{-\Delta x_{ij}}{D^2_{ij}}$$

$$\frac{\partial F}{\partial x_k} = \frac{-\Delta y_{ik}}{D^2_{ik}}$$

$$\frac{\partial F}{\partial y_k} = \frac{\Delta x_{ik}}{D^2_{ik}}$$

$\Delta y_{ij} = y_j - y_i$

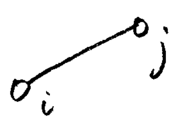
$\Delta y_{ik} = y_k - y_i$

$\Delta x_{ij} = x_j - x_i$

$\Delta x_{ik} = x_k - x_i$

$$F_\theta = \theta_{ijk} - \tan^{-1}\left(\frac{\Delta x_{ik}}{\Delta y_{ik}}\right) + \tan^{-1}\left(\frac{\Delta x_{ij}}{\Delta y_{ij}}\right)$$

use atan2$(\Delta x, \Delta y)$
then force
$[\text{atan2}(\Delta x_{ik}, \Delta y_{ik}) - \text{atan2}(\Delta x_{ij}, \Delta y_{ij})]$
to be positive by adding
2π if necessary



$$F_d = d_{ij} - \left[\Delta x_{ij}^2 + \Delta y_{ij}^2\right]^{1/2}$$

$$\frac{\partial F}{\partial x_i} = \frac{\Delta x_{ij}}{D_{ij}}$$

$$\frac{\partial F}{\partial y_i} = \frac{\Delta y_{ij}}{D_{ij}}$$

$$\frac{\partial F}{\partial x_j} = \frac{-\Delta x_{ij}}{D_{ij}}$$

$$\frac{\partial F}{\partial y_j} = \frac{-\Delta y_{ij}}{D_{ij}}$$

$$D_{ij} = \left[\Delta x_{ij}^2 + \Delta y_{ij}^2\right]^{1/2}$$

$\Delta x_{ij} = x_j - x_i$

$\Delta y_{ij} = y_j - y_i$

I don't show final coordinates because they depend on choice of minimal constraints!

$$W = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & 3.6e{+}5 & & & & & \\ & & & & 3.6e{+}5 & & & & \\ & & & & & 3.6e{+}5 & & & \\ & & & & & & 3.6e{+}5 & & \\ & & & & & & & 3.6e{+}5 & \\ & & & & & & & & 3.6e{+}5 \end{bmatrix}$$

for $\sigma_0^2 = (0.2)^2$

$\sigma_d = 0.2$

$\sigma_a = 0.019^\circ$

$= 0.000331 \; R$

```
% hw4_1_sol.m  10-oct-2011
% 2d network with MC
% syntax of my function calls & returned values
% function [Dij,F,dFdxi,dFdyi,dFdxj,dFdyj]=distance2d(d,i,j,X,Y)
% function [ac,Fa,dFdxi,dFdyi,dFdxj,dFdyj,dFdxk,dFdyk] =
angle2d(a,i,j,k,X,Y)

X=[10332;10044;10426;10637];
Y=[5054;5396;5612;5328];

% fix x,y point 1, and y point 4
% remain free unkowns: x,y point 2, x,y point 3, x point 4
% map coord components to unknown vector
% [ 1  2  3  4  5  6  7  8 ]
% [ -  -  1  2  3  4  5  - ]
map=[0;0;1;2;3;4;5;0];
n=9;
n0=5;
r=4;
u=n0;

dr=180/pi;
l=[447.49;439.78;596.94;97.003/dr;88.249/dr;36.045/dr;79.287/dr;48.521
/dr;95.475/dr];

sd2=(0.2)^2;
s02=sd2;
sa2=(0.019/dr)^2;
wd=s02/sd2;
wa=s02/sa2;
W=diag([wd wd wd wa wa wa wa wa wa]);

niter=0;
keep_going=1;
while((keep_going == 1) & (niter < 10))
  B=zeros(n,u);
  f=zeros(n,1);
  % do all of the distance measurements
  ii=[1;2;2];
  jj=[2;3;4];
  for count=1:3
    obs=count;
    i=ii(count);
    j=jj(count);
    [Dij,F,dFdxi,dFdyi,dFdxj,dFdyj]=distance2d(l(obs),i,j,X,Y);
    idx=map((i-1)*2+1);
    if(idx ~= 0)
      B(obs,idx)=dFdxi;
      end
    idx=map((i-1)*2+2);
    if(idx ~= 0)
      B(obs,idx)=dFdyi;
      end
    idx=map((j-1)*2+1);
    if(idx ~= 0)
      B(obs,idx)=dFdxj;
```

```
      end
    idx=map((j-1)*2+2);
    if(idx ~= 0)
      B(obs,idx)=dFdyj;
      end
    f(obs)=-F;
    end

  % next do all of the angle measurements
  ii=[3;1;2;2;4;4];
  jj=[4;2;3;3;1;1];
  kk=[2;4;4;1;2;3];
  for count=1:6
    obs=3 + count;
    i=ii(count);
    j=jj(count);
    k=kk(count);

[ac,Fa,dFdxi,dFdyi,dFdxj,dFdyj,dFdxk,dFdyk]=angle2d(l(obs),i,j,k,X,Y);
    idx=map((i-1)*2+1);
    if(idx ~= 0)
      B(obs,idx)=dFdxi;
      end
    idx=map((i-1)*2+2);
    if(idx ~= 0)
      B(obs,idx)=dFdyi;
      end
    idx=map((j-1)*2+1);
    if(idx ~= 0)
      B(obs,idx)=dFdxj;
      end
    idx=map((j-1)*2+2);
    if(idx ~= 0)
      B(obs,idx)=dFdyj;
      end
    idx=map((k-1)*2+1);
    if(idx ~= 0)
      B(obs,idx)=dFdxk;
      end
    idx=map((k-1)*2+2);
    if(idx ~= 0)
      B(obs,idx)=dFdyk;
      end
    f(obs)=-Fa;
    end

  N=B'*W*B;
  t=B'*W*f;
  del=inv(N)*t
  X(2)=X(2) + del(1);
  Y(2)=Y(2) + del(2);
  X(3)=X(3) + del(3);
  Y(3)=Y(3) + del(4);
  X(4)=X(4) + del(5);
  niter=niter+1;
  if(all(abs(del) < 0.000001))
```

```
    keep_going=0;
    end
  end

if(keep_going == 1)
  disp('no convergence');
else
  disp('we have converged');
  v=f - B*del
  disp('angle residuals in degrees');
  vd=zeros(6,1);
  for i=1:6
    vd(i)=v(i+3)*dr;
    end
  vd
  lhat=l + v
  disp('adjusted angles in degrees');
  lha=zeros(6,1);
  for i=1:6
    lha(i)=l(i+3) + v(i+3);
    end
  lha=lha*dr;
  lha
  end
```

```
% distance2d.m  30-oct-08
% compute distance condition equation
function [Dij,F,dFdxi,dFdyi,dFdxj,dFdyj]=distance2d(d,i,j,X,Y)
xi=X(i);
yi=Y(i);
xj=X(j);
yj=Y(j);
Dij=sqrt((xj-xi)^2 + (yj-yi)^2);
dFdxi=(xj-xi)/Dij;
dFdyi=(yj-yi)/Dij;
dFdxj=-dFdxi;
dFdyj=-dFdyi;
F=d - Dij;
```

```
% angle2d.m  3-nov-08
function [ac,Fa,dFdxi,dFdyi,dFdxj,dFdyj,dFdxk,dFdyk] =
angle2d(a,i,j,k,X,Y)
xi=X(i);
yi=Y(i);
xj=X(j);
yj=Y(j);
xk=X(k);
yk=Y(k);
Dij_sq=(xj-xi)^2 + (yj-yi)^2;
Dik_sq=(xk-xi)^2 + (yk-yi)^2;
dFdxi= (yk-yi)/Dik_sq - (yj-yi)/Dij_sq;
dFdyi=-(xk-xi)/Dik_sq + (xj-xi)/Dij_sq;
dFdxj=(yj-yi)/Dij_sq;
dFdyj=-(xj-xi)/Dij_sq;
dFdxk=-(yk-yi)/Dik_sq;
dFdyk=(xk-xi)/Dik_sq;
ac=atan2(xk-xi,yk-yi) - atan2(xj-xi,yj-yi);
if(ac < 0)
  ac=ac + 2*pi;
  end

% ac
% degrad=180/pi;
% ac*degrad

Fa=a - ac;
```

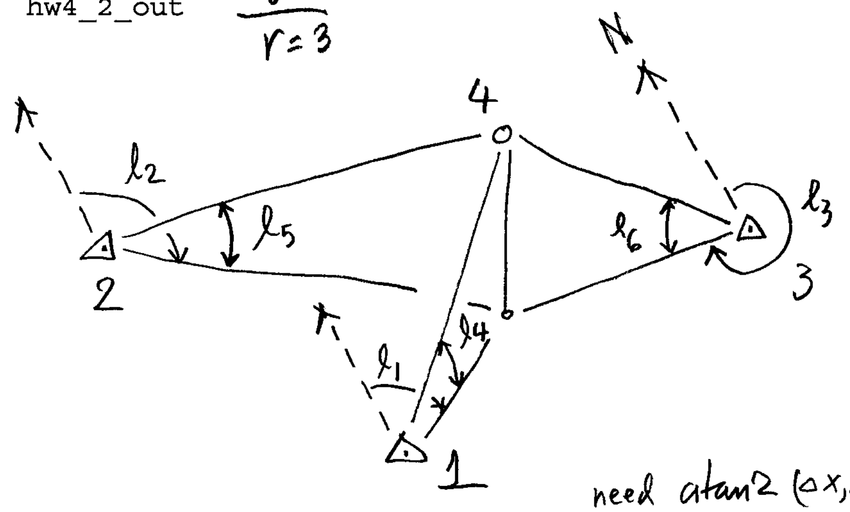
4-2

hw4_2_out

$n=6 \quad u=3 ; \; x_4, y_4, z_4$

$n_0 = 3$

$r = 3$




```
hw4_2_sol
del =
      0.35306
      0.27652
     -0.18357
del =
    0.0025442
    0.0011526
  -0.00071842
del =
  7.4368e-007
   4.987e-007
   2.787e-007
del =
  4.2007e-010
  1.2722e-010
  8.6505e-011
del =
  1.8406e-013
  8.2803e-014
  6.2821e-014
we have converged
point 4
ans =
       96.356       587.48       17.616
v =
   0.00013506
    3.292e-006
  5.1716e-005
  -0.00030556
   0.00052647
  -0.00023114
angle residuals in degrees
vd =
    0.0077383
   0.00018861
    0.0029631
    -0.017507
     0.030164
    -0.013243
lhat =
      0.77419
       1.8049
       4.3719
      0.24228
      0.21127
      0.23532
adjusted angles in degrees
lha =
       44.358
       103.42
       250.49
       13.881
       12.105
       13.483
diary off
```

note iterations

need atan2 $(\Delta x, \Delta y)$ + test for neg & 2π

azimuth

$$F_\alpha = \alpha_{ij} - \tan^{-1}\left(\frac{\Delta x_{ij}}{\Delta y_{ij}}\right)$$

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

$$\frac{\partial F_\alpha}{\partial x_j} = \frac{-\Delta y}{D_{ij}^2}, \quad \frac{\partial F_\alpha}{\partial y_j} = \frac{\Delta x_{ij}}{D_{ij}^2}$$

$$\Delta x_{ij} = x_j - x_i, \quad \Delta y_{ij} = y_j - y_i$$

Vertical Angle

$$F_\beta = \beta_{ij} - \tan^{-1}\left(\frac{\Delta z}{[\Delta x^2 + \Delta y^2]^{1/2}}\right)$$

$$\frac{\partial F_\beta}{\partial x_i} = \frac{-\Delta z \cdot \Delta x}{D^3 + D \cdot \Delta z}, \quad \frac{\partial F_\beta}{\partial x_j} = \frac{+\Delta z \cdot \Delta x}{D^3 + D \cdot \Delta z}$$

$$\frac{\partial F_\beta}{\partial y_i} = \frac{-\Delta z \cdot \Delta y}{D^3 + D \cdot \Delta z}, \quad \frac{\partial F_\beta}{\partial y_j} = \frac{+\Delta z \cdot \Delta y}{D^3 + D \cdot \Delta z}$$

$$\frac{\partial F_\beta}{\partial z_i} = \frac{D}{D^2 + \Delta z^2}, \quad \frac{\partial F_\beta}{\partial z_j} = \frac{-D}{D^2 + \Delta z^2}$$

```
% hw4_2_sol.m  17-oct-2011
% determine unknown 3D point by azimuth and vertical angle
% observations (from fixed control points)
% function calls and returned arguments
% function [ac Fa dFdxi dFdyi dFdxj dFdyj] = azimuth2d(a,i,j,X,Y)
% function [ac Fa dFdxi dFdyi dFdzi dFdxj dFdyj dFdzj] =
vert_angle3d(a,i,j,X,Y,Z)

n=6;
n0=3;
r=3;
u=n0;

% 1,2,3 control points, 4 unknown point appx coords from sketch
X=[60.100;40.500;160.500;96.0];
Y=[550.400;600.800;610.200;587.2];
Z=[4.800;5.300;1.300;17.8];

dr=180/pi;

% equally weighted, W=I_6
W=eye(6);

s02=(0.03/dr)^2;
l=[44.350/dr;103.415/dr;250.491/dr;13.899/dr;12.075/dr;13.496/dr];
%l=[44.350/dr;103.415/dr;250.491/dr;13.5/dr;12.075/dr;13.496/dr];

niter=0;
keep_going=1;
while((keep_going == 1) & (niter < 10))
  B=zeros(n,u);
  f=zeros(n,1);
  % do all of the azimuth measurements
  ii=[1;2;3];
  jj=[4;4;4];
  for count=1:3
    obs=count;
    i=ii(count);
    j=jj(count);
    [ac Fa dFdxi dFdyi dFdxj dFdyj] = azimuth2d(l(obs),i,j,X,Y);
    B(obs,1)=dFdxj;
    B(obs,2)=dFdyj;
    f(obs)=-Fa;
    end
  % next do the vertical angle measurements
  ii=[1;2;3];
  jj=[4;4;4];
  for count=1:3
    obs=3 + count;
    i=ii(count);
    j=jj(count);
    [ac Fa dFdxi dFdyi dFdzi dFdxj dFdyj dFdzj] = ...
       vert_angle3d(l(obs),i,j,X,Y,Z);
    B(obs,1)=dFdxj;
    B(obs,2)=dFdyj;
    B(obs,3)=dFdzj;
```

```
    f(obs)=-Fa;
    end

  N=B'*W*B;
  t=B'*W*f;
  del=inv(N)*t
  X(4)=X(4)+del(1);
  Y(4)=Y(4)+del(2);
  Z(4)=Z(4)+del(3);
  niter=niter+1;
  if(all(abs(del) < 0.0000000001))
    keep_going=0;
    end
  end

if(keep_going == 1)
  disp('no convergence');
else
  disp('we have converged');
  disp('point 4');
  [X(4) Y(4) Z(4)]
  v=f - B*del
  disp('angle residuals in degrees');
  vd=zeros(6,1);
  for i=1:6
    vd(i)=v(i)*dr;
    end
  vd
  lhat=l + v
  disp('adjusted angles in degrees');
  lha=zeros(6,1);
  for i=1:6
    lha(i)=l(i) + v(i);
    end
  lha=lha*dr;
  lha
  end
```

```
% azimuth2d.m  04-oct-2011
function [ac Fa dFdxi dFdyi dFdxj dFdyj] = azimuth2d(a,i,j,X,Y)
xi=X(i);
yi=Y(i);
xj=X(j);
yj=Y(j);
dx=xj-xi;
dy=yj-yi;
Dij_sq=(dx)^2 + (dy)^2;
dFdxi=dy/Dij_sq;
dFdyi=-dx/Dij_sq;
dFdxj=-dFdxi;
dFdyj=-dFdyi;
ac=atan2(xj-xi,yj-yi);
if(ac < 0)
  ac=ac + 2*pi;
  end

% ac
% degrad=180/pi;
% ac*degrad

Fa=a - ac;
```

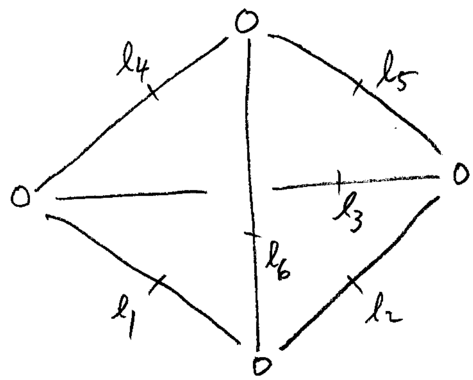
```
% vert_angle3d.m  4-oct-2011

function [ac Fa dFdxi dFdyi dFdzi dFdxj dFdyj dFdzj] =
vert_angle3d(a,i,j,X,Y,Z)
xi=X(i);
yi=Y(i);
zi=Z(i);
xj=X(j);
yj=Y(j);
zj=Z(j);
dx=xj-xi;
dy=yj-yi;
dz=zj-zi;

DHij_sq=(xj-xi)^2 + (yj-yi)^2;
DHij=sqrt(DHij_sq);
denxy=DHij^3 + DHij*dz^2;
denz=DHij^2 + dz^2;
dFdxi=-dz*dx/denxy;
dFdyi=-dz*dy/denxy;
dFdzi=DHij/denz;
dFdxj=-dFdxi;
dFdyj=-dFdyi;
dFdzj=-dFdzi;
ac=atan((zj-zi)/DHij);

% ac
% degrad=180/pi;
% ac*degrad

Fa=a - ac;
```

3. 

"tetrahedron"

(i) (a) to determine size + shape, redundancy = 0

(b) # MC = 6 (3, 2, 1)

(c) # free coordinates = 6

(ii) now observe all VA's & all HA's,
how many MC's ?
2 orientation angles have been determined by VA's, HA's
⇒ # MC = 6 − 2 = 4