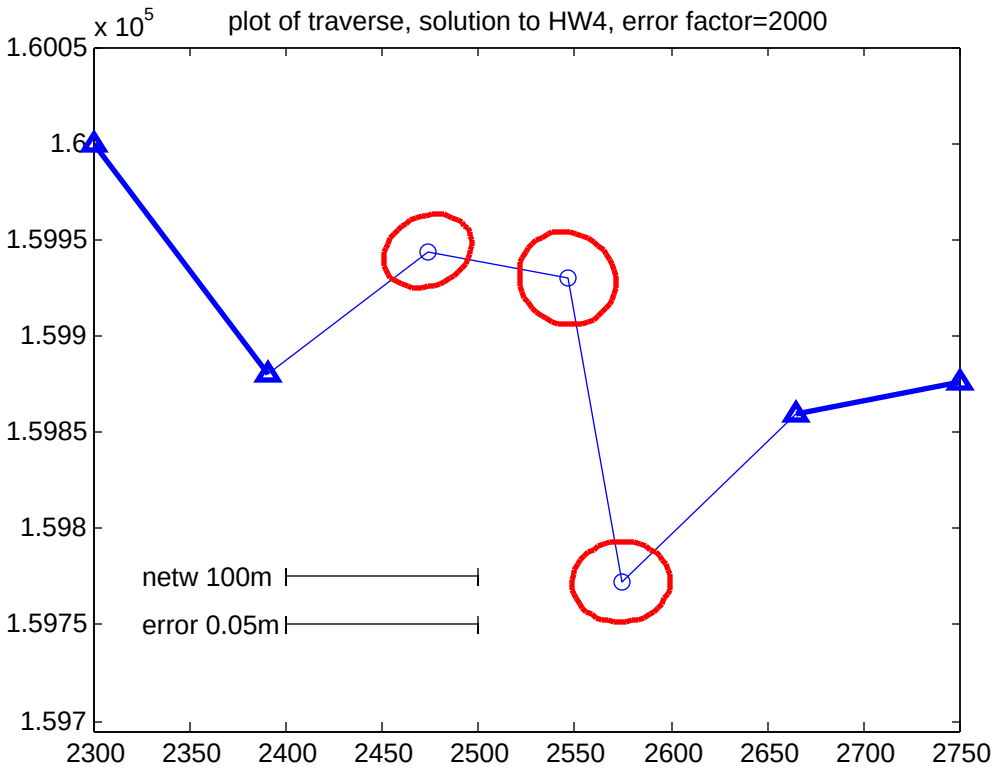
plot of traverse, solution to HW4, error factor=2000
x 10^5
1.6005
1.6
1.5995
1.599
1.5985
1.598
1.5975
1.597
2300
2350
2400
2450
2500
2550
2600
2650
2700
2750
netw 100m
error 0.05m

```
trav_sola
iteration 1 B,f,W
B =
  Columns 1 through 3
      -0.00595659549392845       0.00751324386955303                         0
        -0.783608572548074        -0.621254863183517                         0
       0.00308206049223754       -0.0201950159358364        0.0028745350016909
          0.97526042930658        -0.221059030642825         -0.97526042930658
        0.0028745350016909        0.0126817720662834       -0.00921407065128748
                         0                         0          0.153985421264896
                         0                         0        0.00633953564959657
                         0                         0                         0
                         0                         0                         0
  Columns 4 through 6
                         0                         0                         0
                         0                         0                         0
        0.0126817720662834                         0                         0
         0.221059030642825                         0                         0
       -0.0136697516480387       0.00633953564959657        0.00098797958175531
        -0.988073119783082        -0.153985421264896          0.988073119783082
       0.00098797958175531       -0.00103964659385001       -0.00680111245307197
                         0          0.738976117294387          0.673731621692579
                         0       -0.00529988905574657        0.00581313287131666
f =
       -0.0159862145969785
        -0.700029722807045
        0.0467494125809234
          2.89653571892151
       -0.0111159282304349
         -4.12908944304789
        0.0052293034220724
           1.1298349301227
       -0.0252417394091617
W =
  Columns 1 through 3
           9802.40723623347                         0                         0
                         0          0.983549129907765                         0
                         0                         0          9802.40723623347
                         0                         0                         0
                         0                         0                         0
                         0                         0                         0
                         0                         0                         0
                         0                         0                         0
                         0                         0                         0
  Columns 4 through 6
                         0                         0                         0
                         0                         0                         0
                         0                         0                         0
           1.09262123488691                         0                         0
                         0          9802.40723623347                         0
                         0                         0         0.826476334403407
                         0                         0                         0
                         0                         0                         0
                         0                         0                         0
  Columns 7 through 9
                         0                         0                         0
                         0                         0                         0
                         0                         0                         0
                         0                         0                         0
                         0                         0                         0
                         0                         0                         0
           9802.40723623347                         0                         0
                         0          0.918681423343967                         0
                         0                         0          9802.40723623347
iter =
     1
del =
           1.59359433455018
         -0.886490836145773
         -0.697272251970203
            2.0493279511656
            2.9745809322137
          -1.58020951106068
iter =
     2
del =
       -0.0170373412299926
        0.00603664692223339
         -0.078487102693296
        -0.0493568092013559
         0.0260942646944579
         0.0226257115957175
iter =
     3
del =
       2.29549584816256e-05
      -1.86440693815366e-05
       2.33433870397731e-05
       1.13842052373171e-05
      -2.25537933093467e-05
       1.64061467724694e-05
```

```
iter =
     4
del =
      1.01448383940791e-10
      2.49458162439234e-11
      5.55922091198879e-10
     -3.67557987912026e-10
     -1.38049616564655e-10
     -5.07259938806001e-11
we have converged
final coordinates
ans =
          2473.57657994838          159944.119527167
ans =
          2546.22426398928          159929.999982526
ans =
          2574.00065264298          159772.442432607
residuals
v =
     -8.62567334809892e-05
       0.00166581688040955
     -8.75681073834697e-05
       0.00107761676738717
     -7.89168108470563e-05
     -0.000783938319398868
     -5.42744658380357e-05
       0.00175803528068985
     -5.81501160287448e-05
lhat =
          89.5080248543776
           104.99866581688
           228.50044020967
          74.0070776167674
          249.007615527634
          159.987216061681
          56.2025568366453
          125.993758035281
          212.802219627662
v-a1 seconds
ans =
         -17.7917284189637
v-a2 seconds
ans =
         -18.0622187028763
v-a3 seconds
ans =
         -16.2777606990068
v-a4 seconds
ans =
         -11.1949121802471
v-a5 seconds
ans =
         -11.9943224159152
test_stat =
          7.73648439733946
cv1 =
         0.215795282623898
cv2 =
          9.34840360449614
pass global test, accept Ho
idx =
     1
cov =
       2.8682139092072e-05      4.87970187305266e-06
      4.87970187305266e-06      2.03249518647964e-05
vec =
         0.908413907371805
         0.418071970949369
vec2 =
        -0.418071970949369
         0.908413907371805
a =
         0.011934327493577
b =
        0.00912457239834439
thetad =
          24.7129226694741
z =
          1.64485362695147
sX,sY
sX =
        0.00535557084651786
sY =
        0.00450832029305775
x(3),z*sX,[x(3)-z*sX x(3)+z*sX]
ans =
          2473.57657994838
ans =
        0.00880913013129046
ans =
          2473.56777081825          2473.58538907851
```

```
y(3),z*sY,[y(3)-z*sY y(3)+z*sY]
ans =
          159944.119527167
ans =
       0.00741552698549497
ans =
           159944.11211164           159944.126942694
idx =
     3
cov =
      3.45544780845194e-05     -3.09200789793584e-06
     -3.09200789793584e-06      3.36632283082461e-05
vec =
         0.755859665859495
        -0.654733659992193
vec2 =
         0.654733659992193
         0.755859665859495
a =
        0.0130944040684329
b =
         0.011945322504188
thetad =
         -40.8994577301087
idx =
     5
cov =
       3.6926927889922e-05       1.3294980903233e-06
       1.3294980903233e-06      2.50409124400036e-05
vec =
         0.993951469190589
         0.109820202576167
vec2 =
        -0.109820202576167
         0.993951469190589
a =
        0.0130664172737441
b =
        0.0107070626413525
thetad =
           6.3049511432491
diary off
```

```
% trav_sola.m 01-dec-2013
% data generated by travgen with rng(4)
% do this one with sigma-dist = 5mm + 10 PPM (as assigned)

% cp1  2300.000   160000.000
% cp2  2390.272   159880.205
% 3    2472       159945
% 4    2547       159928
% 5    2571       159774
% cp6  2664.940   159859.646
% cp7  2750.342   159876.246

% d23  104.997
% d34   74.006
% d45  159.988
% d56  125.992

% a213   89.5081     89-30-29.2
% a324  228.5003    228-30-01.9
% a435  249.0077    249-00-27.7
% a546   56.2026     56-12-09.4
% a657  212.8023    212-48-08.2


degrad=180/pi;
x=[2300;2390.272;2472;2547;2571;2664.940;2750.342];
y=[160000;159880.205;159945;159928;159774;159859.646;159876.246];

a1d=89;
a1m=30;
a1s=29.2;
a2d=228;
a2m=30;
a2s=1.9;
a3d=249;
a3m=0;
a3s=27.7;
a4d=56;
a4m=12;
a4s=9.4;
a5d=212;
a5m=48;
a5s=8.2;

a1=a1d + a1m/60 + a1s/3600;
a2=a2d + a2m/60 + a2s/3600;
a3=a3d + a3m/60 + a3s/3600;
a4=a4d + a4m/60 + a4s/3600;
a5=a5d + a5m/60 + a5s/3600;

d1=104.997;
d2=74.006;
d3=159.988;
d4=125.992;

l=[a1; d1; a2; d2; a3; d3; a4; d4; a5];
siga=(12.5/3600)/degrad;
sigd1=0.005 + 10*d1/1e6;
sigd2=0.005 + 10*d2/1e6;
sigd3=0.005 + 10*d3/1e6;
sigd4=0.005 + 10*d4/1e6;
sig0=0.006;
wa=sig0^2/siga^2;
wd1=sig0^2/sigd1^2;
wd2=sig0^2/sigd2^2;
wd3=sig0^2/sigd3^2;
wd4=sig0^2/sigd4^2;
wdiag=[wa wd1 wa wd2 wa wd3 wa wd4 wa];
W=diag(wdiag);


n=9;
n0=6;
r=n-n0;
u=n0;
B=zeros(n,u);
f=zeros(n,1);
max_iter=10;
iter=1;
keep_going=1;
while keep_going == 1

  % l1 - a1 - a213 - angle equation

  degree=a1d;
  minute=a1m;
  second=a1s;
  at=2;
  from=1;
  to=3;
  [b,F,comp_obs]=ang(x,y,at,from,to,degree,minute,second);
```

```
B(1,1)=b(5);
B(1,2)=b(6);
f(1)=-F;

% l2 - d1 - d23 - distance equation

obs=d1;
at=2;
to=3;
[b,F,comp_obs]=dist2d(x,y,at,to,obs);
B(2,1)=b(3);
B(2,2)=b(4);
f(2)=-F;

% l3 - a2 - a324 - angle equation

degree=a2d;
minute=a2m;
second=a2s;
at=3;
from=2;
to=4;
[b,F,comp_obs]=ang(x,y,at,from,to,degree,minute,second);
B(3,1)=b(1);
B(3,2)=b(2);
B(3,3)=b(5);
B(3,4)=b(6);
f(3)=-F;

% l4 - d2 - d34 - distance equation
obs=d2;
at=3;
to=4;
[b,F,comp_obs]=dist2d(x,y,at,to,obs);
B(4,1)=b(1);
B(4,2)=b(2);
B(4,3)=b(3);
B(4,4)=b(4);
f(4)=-F;

% l5 - a3 - a435 - angle equation
degree=a3d;
minute=a3m;
second=a3s;
at=4;
from=3;
to=5;
[b,F,comp_obs]=ang(x,y,at,from,to,degree,minute,second);
B(5,1)=b(3);
B(5,2)=b(4);
B(5,3)=b(1);
B(5,4)=b(2);
B(5,5)=b(5);
B(5,6)=b(6);
f(5)=-F;

% l6 - d3 - d45 - distance equation
obs=d3;
at=4;
to=5;
[b,F,comp_obs]=dist2d(x,y,at,to,obs);
B(6,3)=b(1);
B(6,4)=b(2);
B(6,5)=b(3);
B(6,6)=b(4);
f(6)=-F;

% l7 - a4 - a546 - angle equation
degree=a4d;
minute=a4m;
second=a4s;
at=5;
from=4;
to=6;
[b,F,comp_obs]=ang(x,y,at,from,to,degree,minute,second);
B(7,3)=b(3);
B(7,4)=b(4);
B(7,5)=b(1);
B(7,6)=b(2);
f(7)=-F;

% l8 - d4 - d56 - distance equation
obs=d4;
at=5;
to=6;
[b,F,comp_obs]=dist2d(x,y,at,to,obs);
B(8,5)=b(1);
B(8,6)=b(2);
f(8)=-F;

% l9 - a5 - a657 - angle equation
```

```
  degree=a5d;
  minute=a5m;
  second=a5s;
  at=6;
  from=5;
  to=7;
  [b,F,comp_obs]=ang(x,y,at,from,to,degree,minute,second);
  B(9,5)=b(3);
  B(9,6)=b(4);
  f(9)=-F;

  if iter == 1
    disp('iteration 1 B,f,W');
    B
    f
    W
    end

  % now solve and update and check convergence

  N=B'*W*B;
  t=B'*W*f;
  iter
  del=inv(N)*t
  x(3)=x(3) + del(1);
  y(3)=y(3) + del(2);
  x(4)=x(4) + del(3);
  y(4)=y(4) + del(4);
  x(5)=x(5) + del(5);
  y(5)=y(5) + del(6);
  iter=iter+1;
  if iter > 10
    keep_going=0;
    disp('too many iterations');
    end
  if all(abs(del) < 0.00001)
    keep_going=0;
    disp('we have converged');
    end

  end;

disp('final coordinates');
[x(3) y(3)]
[x(4) y(4)]
[x(5) y(5)]
disp('residuals');
v=f - B*del
lhat=l + v
disp('v-a1 seconds');
v(1)*degrad*3600
disp('v-a2 seconds');
v(3)*degrad*3600
disp('v-a3 seconds');
v(5)*degrad*3600
disp('v-a4 seconds');
v(7)*degrad*3600
disp('v-a5 seconds');
v(9)*degrad*3600

test_stat=v'*W*v/sig0^2
cv1=icdf('chi2',0.025,r)
cv2=icdf('chi2',1-0.025,r)
disp('pass global test, accept Ho');

plot(x,y);
axis equal
hold on
plot(x(1:2),y(1:2),'linewidth',2);
plot(x(6:7),y(6:7),'linewidth',2);
plot([x(1) x(2) x(6) x(7)],[y(1) y(2) y(6) y(7)],'b^','linewidth',2);
plot(x(3:5),y(3:5),'bo');

Qdd=inv(N);
Sdd=sig0^2*Qdd;
for i=1:3
  idx=(i-1)*2 + 1
  cov=Sdd(idx:idx+1,idx:idx+1)
  pndx=i+2;
  plot_conf_ell(cov,1,r,0.90,x(pndx),y(pndx),2000);
  title('plot of traverse, solution to HW4, error factor=2000');
  axis equal
  if(i==1)
    P=0.90;
    z=norminv((P+1)/2,0,1)
    disp('sX,sY');
    sX=sqrt(cov(1,1))
    sY=sqrt(cov(2,2))
    disp('x(3),z*sX,[x(3)-z*sX x(3)+z*sX]');
    x(3)
```

```
    z*sX
    [x(3)-z*sX x(3)+z*sX]
    disp('y(3),z*sY,[y(3)-z*sY y(3)+z*sY]');
    y(3)
    z*sY
    [y(3)-z*sY y(3)+z*sY]
    end
  end
scale_bar(2325,159775,'netw 100m',100,75);
scale_bar(2325,159750,'error 0.05m',0.05*2000,75);
```

```
% 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;
```

```
% 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;
```

```
% plot_conf_ell.m  28-jan-09
% plot confidence ellipse from 2x2 covariance matrix
% cov = 2x2 covariance matrix
% global 1=use chi2 and z, 0= use F amd t
% r=redundancy
% prob=probability (enter as 0.90 not 90 as in 90%)
% x,y = point loc
% factr = enlargement factor for errors

function result=plot_conf_ell(cov,glob,r,prob,x,y,factr)

alpha=1-prob;
ALL_ZEROS=1;
ONLY_X=2;
ONLY_Y=3;
RANK1=4;
RANK2=5;
cs=0;
if((abs(cov(1,1)) < 1.0e-14) & (abs(cov(2,2)) < 1.0e-14))
  cs=ALL_ZEROS;
elseif((abs(cov(1,1)) > 1.0e-14) & (abs(cov(2,2)) < 1.0e-14))
  cs=ONLY_X;
elseif((abs(cov(1,1)) < 1.0e-14) & (abs(cov(2,2)) > 1.0e-14))
  cs=ONLY_Y;
else
  [V,D]=eig(cov);
  % order by magnitude, largest first
  if(D(2,2) > D(1,1))
    temp=D(1,1);
    D(1,1)=D(2,2);
    D(2,2)=temp;
    tempv=V(:,1);
    V(:,1)=V(:,2);
    V(:,2)=tempv;
    end
  if((D(2,2) / D(1,1)) > 1.0e13)
    cs=RANK1;
  else
    cs=RANK2;
    end
  end

% disp('cs = ');
% cs
switch cs
  case ALL_ZEROS
    % plot a point
    plot(x,y,'r.','linewidth',2);
  case ONLY_X
    % plot a horiz line
    sigx=sqrt(cov(1,1));
    if (glob == 1)
      z=norminv(1-alpha/2,0,1);
      scl=z*factr;
    else
      t=tinv(1-alpha/2,r);
      scl=t*factr;
      end
    a=scl*sigx;
    plot([x-a x+a],[y y],'r-','linewidth',2);
  case ONLY_Y
    % plot a vertical line
    sigx=sqrt(cov(2,2));
    if (glob == 1)
      z=norminv(1-alpha/2,0,1);
      scl=z*factr;
    else
      t=tinv(1-alpha/2,r);
      scl=t*factr;
      end
    a=scl*sigx;
    plot([x x],[y-a y+a],'r-','linewidth',2);
  case RANK1
    % plot the tilted line
    vec=V(:,1);
    if(glob == 1)
      a=sqrt(chi2inv(prob,2)*D(1,1));
      a=a*factr;
    else
      a=sqrt(2*finv(prob,2,r)*D(1,1));
      a=a*factr;
      end
    plot([x-vec(1)*a x+vec(1)*a],[y-vec(2)*a y+vec(2)*a],'r-','linewidth',2);
  case RANK2
    % ok plot the ellipse
    vec=V(:,1)
    vec2=V(:,2)
    if(glob == 1)
      a=sqrt(chi2inv(prob,2)*D(1,1))
      a=a*factr;
      b=sqrt(chi2inv(prob,2)*D(2,2))
```

```
    b=b*factr;
  else
    a=sqrt(finv(prob,2,r)*D(1,1))
    a=a*factr;
    b=sqrt(finv(prob,2,r)*D(2,2))
    b=b*factr;
    end
  theta=atan2(vec(2),vec(1));
  thetad=theta*(180/pi)
  draw_ell(x,y,a,b,theta);
end
```

```
% draw_ell.m  22-oct-08
% function to draw ellipse

function result=draw_ell(xorg,yorg,a,b,theta)

th=theta;
x0=a;
y0=0;
nseg=50;
dalpha=2*pi/nseg;
for i=1:nseg
  alpha=i*dalpha;
  x1=a*cos(alpha);
  y1=b*sin(alpha);
  px0=xorg + cos(th)*x0 - sin(th)*y0;
  py0=yorg + sin(th)*x0 + cos(th)*y0;
  px1=xorg + cos(th)*x1 - sin(th)*y1;
  py1=yorg + sin(th)*x1 + cos(th)*y1;
  plot([px0 px1],[py0 py1],'r-','linewidth',2);
  if(i == 1)
    hold on
    end
  x0=x1;
  y0=y1;
  end
result=0;
```

```
% scale_bar.m  26-nov-2013
% draw & label scale bar
% stx: startx of scale bar annotation
% sty: starty of scale bar annotation
% lab: text label at front of annotation
% len: length of scale bar
% strl: offset from beg. of string to scale bar

function res=scale_bar(stx,sty,lab,len,strl)

offs=len/20;
gx=stx + strl;
text(stx,sty,lab);

x=[gx;gx;gx;gx;gx+len;gx+len;gx+len];
y=[sty;sty+offs;sty-offs;sty;sty;sty+offs;sty-offs];
plot(x,y,'k-');
```