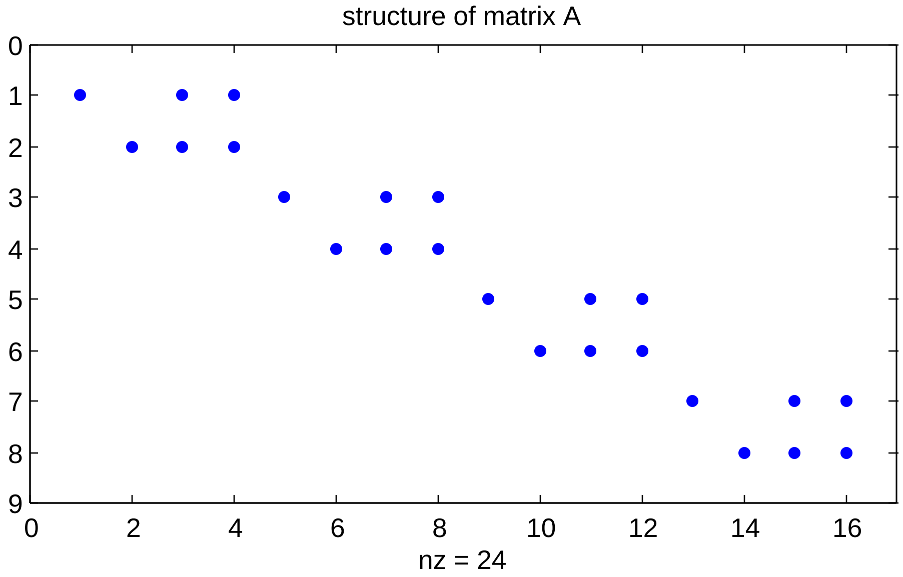
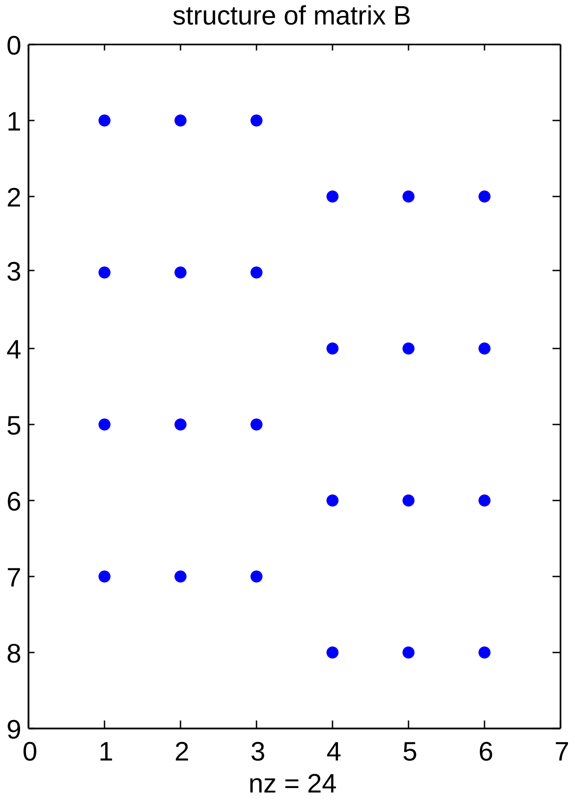
```
% ct1.m 1-dec-08
% coordinate transf. hw5 #1 GLS

x=[17.0;51.5;54.5;88.5];
y=[40.0;19.5;102.5;83.0];
X=[10.05;20.10;9.95;20.15];
Y=[9.90;10.04;19.85;20.08];
l=zeros(16,1);
w=zeros(16,1);
s=0.7;
S=0.2;
% let sigma-naught-squared = 1
for i=1:4
  l((i-1)*4 + 1)=x(i);
  l((i-1)*4 + 2)=y(i);
  l((i-1)*4 + 3)=X(i);
  l((i-1)*4 + 4)=Y(i);
  w((i-1)*4 + 1)=1/s^2;
  w((i-1)*4 + 2)=1/s^2;
  w((i-1)*4 + 3)=1/S^2;
  w((i-1)*4 + 4)=1/S^2;
  end
l0=l;
W=diag(w);

% solve linear problem for parameter approx

B=[-1 -X(1) -Y(1)  0    0     0;
    0    0     0  -1 -X(1) -Y(1);
   -1 -X(2) -Y(2)  0    0     0;
    0    0     0  -1 -X(2) -Y(2);
   -1 -X(3) -Y(3)  0    0     0;
    0    0     0  -1 -X(3) -Y(3)];
f=[-x(1);-y(1);-x(2);-y(2);-x(3);-y(3)];
par=inv(B'*B)*B'*f

% ok now the GLS problem

npt=4
n=16
n0=6 + npt*2
r=n-n0
u=6
c=r+u

keep_going=1;
iter=0;
while (keep_going == 1)
  iter=iter+1
  for i=1:npt
    x(i)=l0((i-1)*4 + 1);
    y(i)=l0((i-1)*4 + 2);
    X(i)=l0((i-1)*4 + 3);
    Y(i)=l0((i-1)*4 + 4);
    end
  a0=par(1);
  a1=par(2);
  a2=par(3);
  b0=par(4);
  b1=par(5);
  b2=par(6);
  B=zeros(c,u);
  A=zeros(c,n);
  F=zeros(c,1);
  for i=1:npt
    indx=(i-1)*4 + 1;
    A((i-1)*2 + 1, indx:indx+3)=[1 0 -a1 -a2];
    A((i-1)*2 + 2, indx:indx+3)=[0 1 -b1 -b2];
    B((i-1)*2 + 1,:)=[-1 -X(i) -Y(i)  0    0     0];
    B((i-1)*2 + 2,:)=[ 0    0     0  -1 -X(i) -Y(i)];
    F((i-1)*2 + 1)=x(i) - a0 - a1*X(i) - a2*Y(i);
    F((i-1)*2 + 2)=y(i) - b0 - b1*X(i) - b2*Y(i);
    end
  %spy(B)
  %title('structure of matrix B');
  %pause
  f=-F - A*(l-l0);
  Q=inv(W);
  Qe=A*Q*A';
  We=inv(Qe);
  N=B'*We*B;
  t=B'*We*f;
  delta=inv(N)*t
  par=par+delta;
  k=We*(f - B*delta);
  v=Q*A'*k;
  if(iter == 1)
    A
    B
    F
    f
    delta
    v
    end
  l0=l+v;
  % check convergence
  if all(abs(delta) < 1.0e-06)
    disp('we have converged')
    keep_going=0;
    end
  if(iter >= 10)
    disp('stopping - too many iterations');
    keep_going=0;
    end;
  end
if(iter < 10)
  v
  test=v'*W*v/1
  cv1=icdf('chi2',0.005,r)
  cv2=icdf('chi2',0.995,r)
  if((test > cv1) & (test < cv2))
    disp('pass global test @ alpha = 0.01');
  else
    disp('fail global test @ alpha = 0.01');
    end
  end
```


structure of matrix A
0
1
2
3
4
5
6
7
8
9
0
2
4
6
8
10
12
14
16
nz = 24

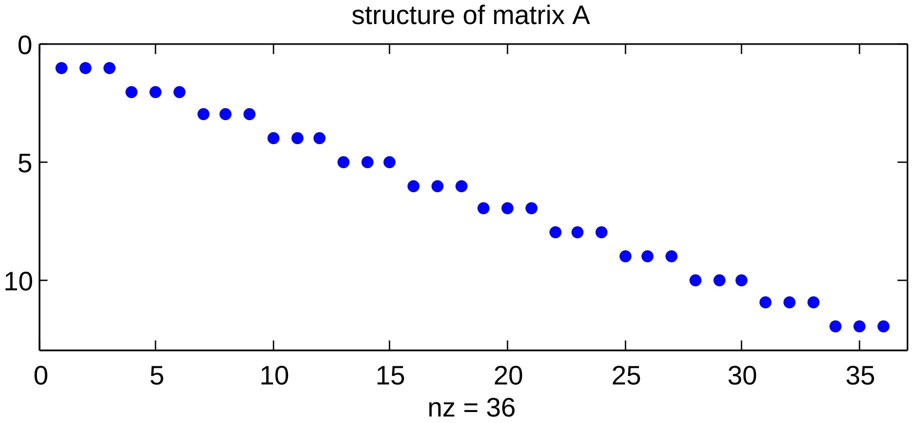

structure of matrix B
0
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
nz = 24

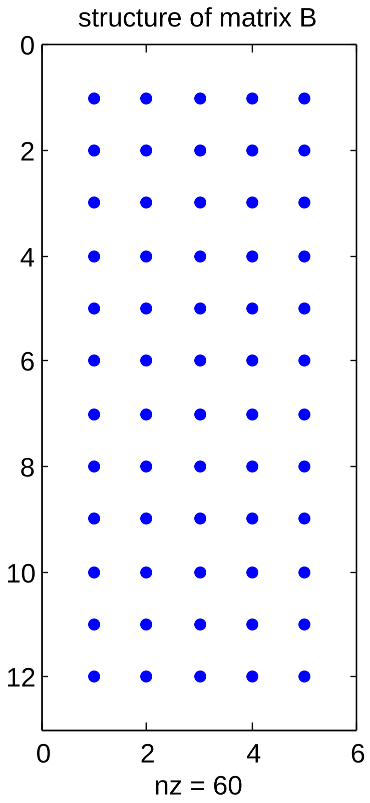
```
ct1
par =
      -54.615
       3.3799
       3.8028
     -0.59789
       -2.127
         6.26
npt =
     4
n =
    16
n0 =
    14
r =
     2
u =
     6
c =
     8
iter =
     1
delta =
        1.672
    -0.065876
    -0.067951
     -0.93642
     0.036894
     0.038057
A =
  Columns 1 through 6
            1            0      -3.3799      -3.8028            0            0
            0            1        2.127        -6.26            0            0
            0            0            0            0            1            0
            0            0            0            0            0            1
            0            0            0            0            0            0
            0            0            0            0            0            0
            0            0            0            0            0            0
            0            0            0            0            0            0
  Columns 7 through 12
            0            0            0            0            0            0
            0            0            0            0            0            0
      -3.3799      -3.8028            0            0            0            0
        2.127        -6.26            0            0            0            0
            0            0            1            0      -3.3799      -3.8028
            0            0            0            1        2.127        -6.26
            0            0            0            0            0            0
            0            0            0            0            0            0
  Columns 13 through 16
            0            0            0            0
            0            0            0            0
            0            0            0            0
            0            0            0            0
            0            0            0            0
            0            0            0            0
            1            0      -3.3799      -3.8028
            0            1        2.127        -6.26
B =
           -1       -10.05         -9.9            0            0            0
            0            0            0           -1       -10.05         -9.9
           -1        -20.1       -10.04            0            0            0
            0            0            0           -1        -20.1       -10.04
           -1        -9.95       -19.85            0            0            0
```

```
            0            0            0           -1        -9.95       -19.85
           -1       -20.15       -20.08            0            0            0
            0            0            0           -1       -20.15       -20.08
F =
  1.4921e-013
  2.1316e-014
 -2.1316e-014
  8.5265e-014
  4.2633e-014
   1.279e-013
      -1.3492
      0.75565
f =
 -1.4921e-013
 -2.1316e-014
  2.1316e-014
 -8.5265e-014
 -4.2633e-014
  -1.279e-013
       1.3492
     -0.75565
delta =
        1.672
    -0.065876
    -0.067951
     -0.93642
     0.036894
     0.038057
v =
      0.14512
    -0.084434
      -0.0547
   -0.0019026
     -0.14387
     0.083706
     0.054228
    0.0018862
     -0.14299
     0.083196
     0.053898
    0.0018747
      0.14174
    -0.082468
    -0.053426
   -0.0018583
iter =
     2
delta =
     -0.01079
   0.00070282
  1.3647e-005
    0.0060433
  -0.00039362
 -7.6429e-006
iter =
     3
delta =
   7.661e-006
  1.9447e-006
 -2.4689e-006
 -4.2906e-006
 -1.0891e-006
  1.3827e-006
```

```
iter =
     4
delta =
 -9.7123e-007
  6.4208e-008
  2.7403e-010
  5.4395e-007
  -3.596e-008
 -1.5348e-010
we have converged
v =
      0.14745
    -0.084138
    -0.054257
   -0.0016992
     -0.14621
     0.083428
       0.0538
    0.0016849
     -0.14842
     0.084691
     0.054614
    0.0017104
      0.14718
    -0.083981
    -0.054156
    -0.001696
test =
      0.52898
cv1 =
     0.010025
cv2 =
       10.597
pass global test @ alpha = 0.01
diary off
```

```
% ellfit.m 11-dec-08
% fit data to rotational ellipsoid by GLS
x=[15.1;5.3;4.4;16.2;19.8;11.0;14.2;4.5;4.7;12.9;0.3;9.5];
y=[25.2;23.7;18.1;18.5;20.2;29.0;22.2;21.7;18.6;15.3;19.9;11.1];
z=[36.2;37.2;37.3;36.9;31.8;33.8;22.1;22.6;22.5;22.5;27.8;25.9];
[mr,mc]=size(x);
npt=mr;
xbar=sum(x)/npt;
ybar=sum(y)/npt;
zbar=sum(z)/npt;
rangex=max(x)-min(x);
rangey=max(y)-min(y);
rangez=max(z)-min(z);
a=0.5*sqrt(rangex^2 + rangey^2);
b=0.5*rangez;
xc=xbar;
yc=ybar;
zc=zbar;
%
% F = ((x-xc)^2 + (y-yc)^2)/a^2 + (z-zc)^2/b^2 - 1 = 0
%
n=npt*3
u=5
n0=u+npt*2
r=n-n0
c=r+u
%
% pack the observations
%
l=zeros(n,1);
for i=1:12
  idx=(i-1)*3 + 1;
  l(idx)=x(i);
  l(idx+1)=y(i);
  l(idx+2)=z(i);
  end
l0=l;
x0=zeros(npt,1);
y0=zeros(npt,1);
z0=zeros(npt,1);
W=eye(n);
Q=inv(W);
keep_going=1;
iter=0;
while(keep_going == 1)
  iter=iter+1
  B=zeros(c,u);
  A=zeros(c,n);
  f=zeros(c,1);
  % unpack the obs back into xyz arrays
  for i=1:npt
    idx=(i-1)*3+1;
    x0(i)=l0(idx);
    y0(i)=l0(idx+1);
    z0(i)=l0(idx+2);
    end
  for i=1:c
    dFda=-((x0(i)-xc)^2 + (y0(i)-yc)^2)*2*a/a^4;
    dFdb=-((z0(i)-zc)^2)*2*b/b^4;
    dFdxc=(1/a^2)*2*(x0(i)-xc)*(-1);
    dFdyc=(1/a^2)*2*(y0(i)-yc)*(-1);
    dFdzc=(1/b^2)*2*(z0(i)-zc)*(-1);
    dFdx=-dFdxc;
    dFdy=-dFdyc;
    dFdz=-dFdzc;
    B(i,:)=[dFda dFdb dFdxc dFdyc dFdzc];
    idx=(i-1)*3+1;
    Asub=[dFdx dFdy dFdz];
    A(i,idx:idx+2)=Asub;
    F=((x0(i)-xc)^2 + (y0(i)-yc)^2)/a^2 + ((z0(i)-zc)^2)/b^2 - 1;
    f(i)= -F - Asub*[x(i)-x0(i); y(i)-y0(i); z(i)-z0(i)];
    end
  %spy(B)
  %title('structure of matrix B');
  %pause
  Qe=A*Q*A';
  We=inv(Qe);
  N=B'*We*B;
  t=B'*We*f;
  del=inv(N)*t
  a=a+del(1);
  b=b+del(2);
  xc=xc+del(3);
  yc=yc+del(4);
  zc=zc+del(5);
  v=Q*A'*We*(f - B*del);
  l0=l+v;
  if(all(abs(del) < 1.0e-06))
    keep_going=0;
    disp('we have converged');
    end
  end

% error prop - assumed W=I so not global test to make
% default to "fail" mode, use F & t

sig0_hat_sqr=v'*W*v/r;
Qdd=inv(N);
Sdd=sig0_hat_sqr*Qdd;

Sxy=Sdd(3:4,3:4);
disp('careful of ordering of lambda 1 & 2');
[V,D]=eig(Sxy);
V
D
Fscale=icdf('f',0.95,2,r);
ax1=sqrt(D(1,1)*2*Fscale)
ax2=sqrt(D(2,2)*2*Fscale)
```


structure of matrix A
0
5
10
0
5
10
15
20
25
30
35
nz = 36


structure of matrix B
0
2
4
6
8
10
12
0
2
4
6
nz = 60

```
ellfit
n =
    36
u =
     5
n0 =
    29
r =
     7
c =
    12
iter =
     1
del =
      -4.9898
       1.0841
      0.29808
     -0.52686
      0.20686
iter =
     2
del =
       1.3217
      0.29306
     -0.09207
      0.15642
     0.076672
iter =
     3
del =
      0.41036
     0.020109
    -0.038118
     0.066816
    0.0056368
iter =
     4
del =
      0.02858
   0.00048965
   -0.0032778
    0.0060938
  3.4125e-005
iter =
     5
del =
   0.00012587
  2.8573e-006
 -1.1058e-006
  2.7987e-005
 -8.6355e-007
iter =
     6
del =
  6.4633e-007
 -6.7656e-007
 -5.8682e-008
  1.8498e-007
 -1.4156e-007
we have converged
careful of ordering of lambda 1 & 2
V =
     -0.99994     0.011099
    -0.011099     -0.99994
D =
   0.00017823            0
            0   0.00033255
ax1 =
     0.041094
ax2 =
     0.056133
diary off
```