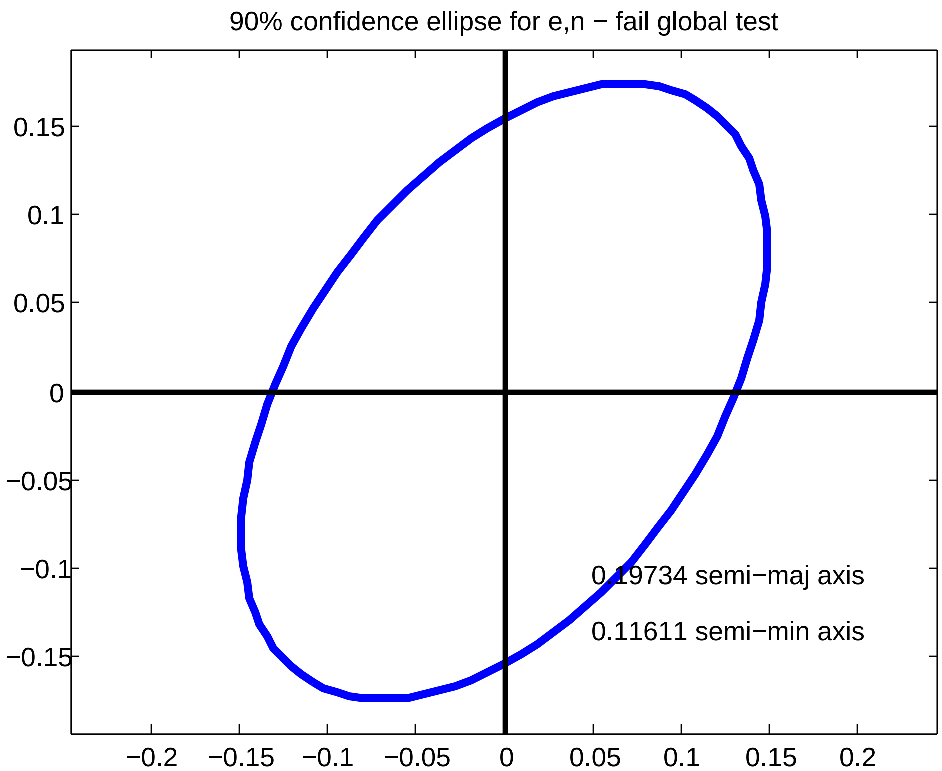
```
gps_nav31
numsat =
     8
sats =
     2
     5
    10
    12
    15
    21
    29
    30
after editing out defective sats
numsat =
     7
ans =
  2.1833e+007        17716       -13232        14327       186.71
  2.0311e+007       3863.7       -17383        19670      -53.325
  2.0489e+007       7237.5       -14735        20642      -27.242
  2.4717e+007      -9332.9       -23015      -9530.3      -196.45
  2.3077e+007       6777.3       -25299      -4115.3      -320.39
  2.4928e+007       -23318      -6190.3        12070      -18.572
  2.1069e+007        -9003       -14746        20274       70.656
condN =
       96.618
disp_del =
            1
     0.056612
   0.00049731
  -0.00061017
   -0.0010735
condN =
       96.618
disp_del =
            2
 -3.6366e-009
 -1.3619e-008
  5.2691e-008
 -9.2354e-008
we have converged
receiver location (km)
ans =
       262.06      -4855.1       4114.4
receiver clock bias (usec)
rdt =
   -0.0010736
residuals (km)
v =
     0.012438
    0.0093152
    0.0087287
     0.054929
    -0.058417
    -0.022755
   -0.0042392
rms =
     0.032256
r =
     3
test_stat =
       11.653
cv1 =
       0.2158
cv2 =
       9.3484
fail global test
convert XYZ to phi,lam,h
phioo =
       0.7056
N =
       6387.1
new_phi =
       0.7056
phioo =
       0.7056
phi
result =
           40
           25
       39.622
lambda
result =
          -86
```

```
         -54
     -37.399
h
h =
     0.17422
J =
     0.99855     0.053898            0
   -0.034952      0.64755      0.76123
    0.041028     -0.76012      0.64849
cov enu
mx =
   0.0020205     0.001102
    0.001102     0.002778
alen =
     0.19734
blen =
     0.11611
theta =
     -2.1907
zfact =
      2.3534
half_width =
     0.23665
width =
     0.47331
diary off
```


90% confidence ellipse for e,n − fail global test
0.15
0.1
0.05
0
−0.05
−0.1
−0.15
−0.2
−0.15
−0.1
−0.05
0
0.05
0.1
0.15
0.2
0.19734 semi−maj axis
0.11611 semi−min axis

```
% gpsnav31.m 8 dec-09
% derived from gpsnav20.m
% derived from gpsnav16.m
% solve gps pseudorange problem for 1 epoch
% adapted from brian yentes 2004 solution
% now 2005 problem from bvg & jen-yu han

% note you need to edit the obs file to replace spaces with
% zeros in satellite number string, and you need to edit the
% data fields to replace spaces with zeros ??? why are
% spaces there ???

% maybe need an epoch = 1,2,3,4 variable for multiple epoch case
fido=fopen('epoch1.txt','rt');

% interpret satellite prn's in the observation file
S=textscan(fido,'%d %d %d %d %d %f %d %s',1);
% need to edit leading zero into single digit sat number fields
str=char(S{8});
ck1=double(str(1));
ck2=double(str(2));
start_char=0;
proceed=0;
if((ck1 >= 48) & (ck1 <= 57))
  start_char=2;
  end
if((ck2 >= 48) & (ck2 <= 57))
  start_char=3;
  end
switch start_char
  case 0
    disp('cannot interpret satellite string');
    proceed=0;
  case 2
    proceed=1;
    numsat=str2num(str(1));
  case 3
    proceed=1;
    numsat=str2num(str(1:2));
    if(numsat > 15)
      disp('too many satellites');
      proceed=0;
      end
  end
sats=zeros(numsat,1);
if(proceed == 1)
  run_char=start_char;
  for i=1:numsat
    sats(i)=str2num(str(run_char+1:run_char+2));
    run_char=run_char+3;
    end
  end

numsat
sats

%disp('pause, press a key to continue');
%pause

S=textscan(fido,'%f %f %f %f',numsat);
c1=S{1};
fclose(fido);
%pause

fids=fopen('epoch1s.txt','rt');
% interpret first line of satellite file
S=textscan(fids,'%s %d %d %d %d %d %f',1);
% interpret the satellite data
S=textscan(fids,'%s %f %f %f %f %f',32);
fclose(fids);
tsat=S{2};
XX=S{3};
YY=S{4};
ZZ=S{5};
tdt=S{6};
Xs=zeros(numsat,1);
Ys=zeros(numsat,1);
Zs=zeros(numsat,1);
dt=zeros(numsat,1);
for i=1:numsat
  % last year this number was 31, this year it is 30
  % seems satellite #25 missing
  for j=1:30
```

```
    if(tsat(j) == sats(i))
      Xs(i)=XX(j);
      Ys(i)=YY(j);
      Zs(i)=ZZ(j);
      dt(i)=tdt(j);
      end
    end
  end
[m,n]=size(Xs);
if(numsat ~= m)
  disp('error in satellite counting');
  pause
  end

% edit out any sats with invalid dt and also corresponding obs
%[c1 Xs Ys Zs dt]
keep_going=1;
while (keep_going==1)
  change=0;
  for i=1:numsat
    if(dt(i) > 999999.0)
      c1(i)=[];
      Xs(i)=[];
      Ys(i)=[];
      Zs(i)=[];
      dt(i)=[];
      change=1;
      break
      end
    end
  if(change == 1)
    numsat=numsat-1;
  else
    keep_going=0;
    end
  end

disp('after editing out defective sats');
numsat
[c1 Xs Ys Zs dt]

nobs=numsat;

%disp('ok check variables');
%pause

npar=4;
n=nobs;
n0=npar;
r=n-n0;
% Xs,Ys,Zs (km), c1 (m), DT(us)
% we solve in km & us

rawpr=c1/1000; % convert to km
DT=dt;
W=eye(nobs);
sigma0=0.025;

% units
% c1 (unrefined pseudorange) is in meters in file
% xs,xs,zs in km
% DT 1e-06 sec, i.e. us or microseconds
c=0.299792458; % km/us (km / u-second)
pr=rawpr + c*DT;
% initial approximations to receiver coords
% found in the observation file header
% convert from given m to km
Xo= 262004.0/1000;
Yo= -4855113.0/1000;
Zo= 4114363.0/1000;
%Xo=0.0;
%Yo=0.0;
%Zo=0.0;
rdt=0.0;
old_phi=9.99e+09;
threshold=1.0e-06;
converged=0;
for iter=1:10
  B=zeros(nobs,npar);
  f=zeros(nobs,1);
  for i=1:nobs
    D=sqrt((Xs(i)-Xo)^2 + (Ys(i)-Yo)^2 + (Zs(i)-Zo)^2);
    B(i,1)=(Xs(i)-Xo)/D;
```

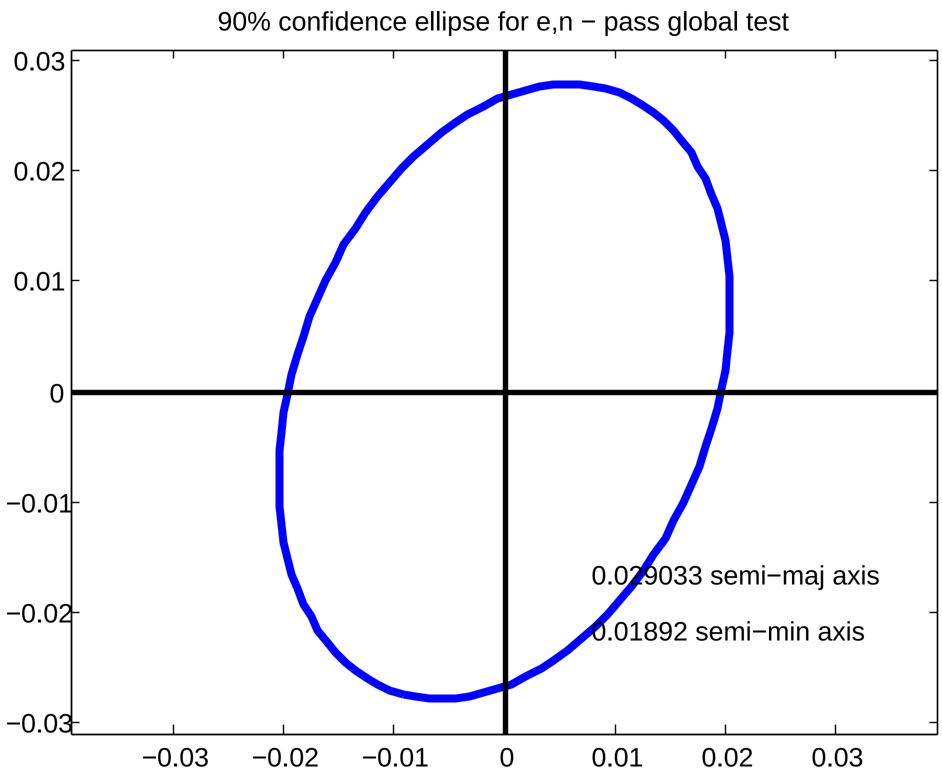
```
    B(i,2)=(Ys(i)-Yo)/D;
    B(i,3)=(Zs(i)-Zo)/D;
    B(i,4)=-c;
    F=pr(i) - D - c*rdt;
    f(i)=-F;
    end
  % B
  % f
  % pause
  % W
  %condJ=cond(B)
  N=B'*W*B;
  condN=cond(N)
  t=B'*W*f;
  Ni=inv(N);
  del=Ni*t;
  Qdd=Ni;
  Xo=Xo + del(1);
  Yo=Yo + del(2);
  Zo=Zo + del(3);
  rdt=rdt + del(4);
  disp_del=[iter; del(1); del(2); del(3); del(4)]
  v=f-B*del;
  phi=v'*W*v;
  if(abs(phi-old_phi)/phi < threshold)
    disp('we have converged');
    converged=1;
    break;
    end

  old_phi=phi;
  end

if(converged == 0)
  disp('we did not converge');
else
  % we converged
  % show results
  disp('receiver location (km)');
  [Xo Yo Zo]
  disp('receiver clock bias (usec)');
  [rdt]
  end
disp('residuals (km)');
v
rms=sqrt(v'*v/nobs)

% make global test and post adjustment statistics

r
test_stat=v'*W*v/(sigma0^2)
cv1=icdf('chi2',0.025,r)
cv2=icdf('chi2',0.975,r)
if((test_stat > cv1) && (test_stat < cv2))
  Sdd=sigma0^2 * Qdd;
  pass=1;
  disp('pass global test');
else
  sigma0_hat_sqr=v'*W*v/r;
  Sdd=sigma0_hat_sqr * Qdd;
  pass=0;
  disp('fail global test');
  end

% ok transform cofactor matrix to ENU from XYZ
% first get lat, lon, h from XYZ by iteration
% remember Xo,Yo,Zo are km

disp('convert XYZ to phi,lam,h');
lam=atan2(Yo,Xo);
a=6378137.0/1000;
f=1/298.257223563;
esqr=2*f-f^2;
e=sqrt(esqr);
phioo=atan(Zo/((1-e^2)*sqrt(Xo^2+Yo^2)))
keep_going=1;
while(keep_going == 1)
  N=a/sqrt(1-esqr*(sin(phioo))^2)
  new_phi=atan((Zo/sqrt(Xo^2 + Yo^2))*(1 + esqr*N*sin(phioo)/Zo))
  if(abs(new_phi - phioo) < 1.0e-06);
    keep_going=0;
    end
  phioo=new_phi
```

```
  end
N=a/sqrt(1-esqr*(sin(phioo))^2);
phi=phioo;
h=sqrt(Xo^2 + Yo^2)/cos(phi) - N;
disp('phi');
result=raddms(phi)
disp('lambda');
result=raddms(lam)
disp('h');
h

% rotate XYZ into enu
% bvg uses "enu" for local cartesian and ENH for map projections
% extract the submatrix for XYZ to transform
M=m1(pi/2 - phi)*m3(lam + pi/2);
J=M;
J
Sdd_enu=J*Sdd(1:3,1:3)*J';
disp('cov enu');
mx=Sdd_enu(1:2,1:2)
sigz_sqr=Sdd_enu(3,3);
[V,D]=eig(mx);
if(D(1,1) < D(2,2))
  tmp=D(1,1);
  D(1,1)=D(2,2);
  D(2,2)=tmp;
  tmp=V(:,1);
  V(:,1)=V(:,2);
  V(:,2)=tmp;
  end

% 90%
P=0.90;
alpha=1-P;
if(pass == 1)
  alen=sqrt(D(1,1)*icdf('chi2',P,2))
  blen=sqrt(D(2,2)*icdf('chi2',P,2))
  theta=atan2(V(2,1),V(1,1))
  plot_ell(0,0,alen,blen,theta);
  title('90% confidence ellipse for e,n - pass global test');
  zfact=icdf('norm',1-alpha/2,0,1)
  half_width=zfact*sqrt(sigz_sqr)
  width=2*half_width
else
  alen=sqrt(D(1,1)*2*icdf('f',P,2,r))
  blen=sqrt(D(2,2)*2*icdf('f',P,2,r))
  theta=atan2(V(2,1),V(1,1))
  plot_ell(0,0,alen,blen,theta);
  title('90% confidence ellipse for e,n - fail global test');
  zfact=icdf('t',1-alpha/2,r)
  half_width=zfact*sqrt(sigz_sqr)
  width=2*half_width
  end
axis equal
```

```
gps_nav32
numsat =
      8
sats =
      2
      5
     10
     12
     15
     21
     29
     30
numsat =
      7
sats =
      2
      5
     10
     15
     21
     29
     30
numsat =
      7
sats =
      2
      5
     10
     15
     21
     29
     30
numsat =
      7
sats =
      2
      5
     10
     15
     18
     21
     29
nobs =
     29
ans =
     29      1
ans =
     29      1
ans =
     29      1
ans =
     29      1
ans =
     29      1
nobs =
     28
threw one out
nobs =
     27
threw one out
nobs =
     26
threw one out
nobs =
     25
threw one out
found no observations to throw out
after editing out defective sats
nobs =
     25
ans =
  2.1833e+007        17716       -13232        14327       186.71
  2.0311e+007       3863.7       -17383        19670      -53.325
  2.0489e+007       7237.5       -14735        20642      -27.242
  2.4717e+007      -9332.9       -23015      -9530.3      -196.45
  2.3077e+007       6777.3       -25299      -4115.3      -320.39
  2.4928e+007       -23318      -6190.3        12070      -18.572
  2.1069e+007        -9003       -14746        20274       70.656
  2.2211e+007        19281       -13244        12097       186.71
  2.0493e+007         5422       -15668        20721      -53.318
  2.0804e+007       8811.1       -12867        21295      -27.243
   2.252e+007         7142       -25502      -1288.1      -320.39
   2.446e+007       -21919      -6622.7        14257      -18.574
```

```
  2.0826e+007      -8069.1       -16776        19050       70.658
  2.2627e+007        20587       -13266         9653       186.71
  2.0741e+007       7134.6       -13964        21415      -53.311
  2.1178e+007        10511       -11043        21574      -27.243
  2.2011e+007         7406       -25415       1561.3      -320.39
  2.3989e+007       -20318        -7172        16210      -18.576
  2.0669e+007      -7292.2       -18703        17502       70.661
  2.3079e+007        21615       -13265       7038.7       186.72
  2.1044e+007       8975.3       -12314        21739      -53.304
  2.1602e+007        12306      -9303.4        21477      -27.244
  2.1564e+007       7604.9       -25027       4383.7      -320.39
  2.3518e+007       -18550      -7855.3        17896      -18.578
  2.0605e+007      -6665.7       -20484        15655       70.664
disp_del =
            1
       267.15
        -5794
         4917
disp_del2 =
       4233.6       4299.2       4319.9       4325.5
disp_del =
            2
      -7.3178
       910.68
      -778.42
disp_del2 =
      -4099.2      -4160.4      -4179.7      -4185.2
disp_del =
            3
       2.2369
       28.184
      -24.133
disp_del2 =
      -134.23      -138.52      -139.88      -140.15
disp_del =
            4
    0.0039968
     0.027462
     -0.02299
disp_del2 =
     -0.12807     -0.13124     -0.13206     -0.13192
disp_del =
            5
  7.8265e-009
  3.7624e-008
 -3.7457e-008
disp_del2 =
 -1.5867e-007 -1.6755e-007 -1.6779e-007 -1.6648e-007
we have converged
receiver location (km)
ans =
       262.07      -4855.1       4114.4
receiver clock biases (usec)
ans =
    -0.019969      0.15356      0.12985   -0.0044039
residuals (km)
v =
   -0.0025678
   0.00033729
   -0.0058786
     0.081199
    -0.039894
    -0.023327
   -0.0098692
     0.010681
     0.010556
    0.0044252
    -0.026023
    -0.009465
    0.0098267
      0.01091
    0.0070048
   0.00073631
    -0.025157
    -0.010363
     0.016869
     0.011286
     0.003103
   -0.0027484
    -0.024094
    -0.011807
      0.02426
rms =
```

```
      0.02247
r =
    18
test_stat =
       20.196
cv1 =
       8.2307
cv2 =
       31.526
pass global test
convert XYZ to phi,lam,h
phioo =
       0.7056
N =
       6387.1
new_phi =
       0.7056
phioo =
       0.7056
phi
result =
           40
           25
       40.578
lambda
result =
          -86
          -54
      -36.888
h
h =
      0.17046
J =
      0.99855       0.0539            0
    -0.034954      0.64755      0.76122
      0.04103     -0.76012      0.64849
alen =
     0.029033
blen =
      0.01892
theta =
      -1.9362
zfact =
       1.6449
half_width =
     0.038612
width =
     0.077223
diary off
```


90% confidence ellipse for e,n − pass global test
0.03
0.02
0.01
0
−0.01
−0.02
−0.03
−0.03
−0.02
−0.01
0
0.01
0.02
0.03
0.029033 semi−maj axis
0.01892 semi−min axis

```
% gpsnav32.m 8-dec-09
% this will be the 4-epoch solution
% derived from gpsnav21.m
% derived from gpsnav16.m
% solve gps pseudorange problem for 1 epoch
% adapted from brian yentes 2004 solution
% now 2005 problem from bvg & jen-yu han

% note you need to edit the obs file to replace spaces with
% zeros in satellite number string, and you need to edit the
% data fields to replace spaces with zeros ??? why are
% spaces there ???

% 4 epochs extracted from NGS/CORS RINEX files
obsfile=['epoch1.txt'; 'epoch2.txt'; 'epoch3.txt'; 'epoch4.txt'];
satfile=['epoch1s.txt'; 'epoch2s.txt'; 'epoch3s.txt'; 'epoch4s.txt'];
nobs=0;

for k=1:4
  % maybe need an epoch = 1,2,3,4 variable for multiple epoch case
  fido=fopen(obsfile(k,:),'rt');

  % interpret satellite prn's in the observation file
  S=textscan(fido,'%d %d %d %d %d %f %d %s',1);
  str=char(S{8});
  ck1=double(str(1));
  ck2=double(str(2));
  start_char=0;
  proceed=0;
  if((ck1 >= 48) & (ck1 <= 57))
    start_char=2;
    end
  if((ck2 >= 48) & (ck2 <= 57))
    start_char=3;
    end
  switch start_char
    case 0
      disp('cannot interpret satellite string');
      proceed=0;
    case 2
      proceed=1;
      numsat=str2num(str(1));
    case 3
      proceed=1;
      numsat=str2num(str(1:2));
      if(numsat > 15)
        disp('too many satellites');
        proceed=0;
        end
    end
  sats=zeros(numsat,1);
  if(proceed == 1)
    run_char=start_char;
    for i=1:numsat
      sats(i)=str2num(str(run_char+1:run_char+2));
      run_char=run_char+3;
      end
    end

  numsat
  sats

  %disp('pause, press a key to continue');
  %pause

  S=textscan(fido,'%f %f %f %f',numsat);
  tc1=S{1};
  fclose(fido);

  fids=fopen(satfile(k,:),'rt');
  % interpret first line of satellite file
  S=textscan(fids,'%s %d %d %d %d %d %f',1);
  % interpret the satellite data
  S=textscan(fids,'%s %f %f %f %f %f',32);
  fclose(fids);
  tsat=S{2};
  XX=S{3};
  YY=S{4};
  ZZ=S{5};
  TT=S{6};
  tXs=zeros(numsat,1);
  tYs=zeros(numsat,1);
  tZs=zeros(numsat,1);
```

```
  tdt=zeros(numsat,1);
  for i=1:numsat
    % last year this number was 31, this year it is 30
    % seems satellite #25 missing
    for j=1:30
      if(tsat(j) == sats(i))
        tXs(i)=XX(j);
        tYs(i)=YY(j);
        tZs(i)=ZZ(j);
        tdt(i)=TT(j);
        end
      end
    end
  [m,n]=size(tXs);
  if(numsat ~= m)
    disp('error in satellite counting');
    pause
    end
  nobs=nobs+numsat;
  tepch=ones(numsat,1);
  tepch=tepch*k;
  % transfer data into big arrays
  if(k == 1)
    epoch=tepch;
    c1=tc1;
    Xs=tXs;
    Ys=tYs;
    Zs=tZs;
    dt=tdt;
  else
    epoch=[epoch; tepch];
    c1=[c1; tc1];
    Xs=[Xs; tXs];
    Ys=[Ys; tYs];
    Zs=[Zs; tZs];
    dt=[dt; tdt];
    end
  end

nobs
size(c1)
size(Xs)
size(Ys)
size(Zs)
size(dt)
pause

% edit out any sats with invalid dt and also corresponding obs
%[c1 Xs Ys Zs dt]
keep_going=1;
while (keep_going==1)
  change=0;
  for i=1:nobs
    if(dt(i) > 999999.0)
      c1(i)=[];
      Xs(i)=[];
      Ys(i)=[];
      Zs(i)=[];
      dt(i)=[];
      epoch(i)=[];
      change=1;
      break
      end
    end
  if(change == 1)
    nobs=nobs-1
    disp('threw one out');
  else
    keep_going=0;
    disp('found no observations to throw out');
    end
  end

disp('after editing out defective sats');
nobs
[c1 Xs Ys Zs dt]


%disp('ok check variables');
%pause

npar=3 + 4;
n=nobs;
```

```
n0=npar;
r=n-n0;
% Xs,Ys,Zs (km), c1 (m), DT(us)
% we solve in km & us
rawpr=c1/1000; % convert to km
DT=dt;
%
W=eye(nobs);
sigma0=0.025;

% units
% c1 (unrefined pseudorange) is in meters in file
% xs,xs,zs in km
% DT 1e-06 sec, i.e. us or microseconds
c=0.299792458; % km/us (km / u-second)
pr=rawpr + c*DT;
% initial approximations to receiver coords
% found in the observation file header
% convert from given m to km
Xo= 262004.0/1000;
Yo= -4855113.0/1000;
Zo= 4114363.0/1000;
Xo=0.0;
Yo=0.0;
Zo=0.0;
rdt=[0.0; 0.0; 0.0; 0.0];
old_phi=9.99e+09;
threshold=1.0e-06;
converged=0;
for iter=1:10
  B=zeros(nobs,npar);
  f=zeros(nobs,1);
  for i=1:nobs
    D=sqrt((Xs(i)-Xo)^2 + (Ys(i)-Yo)^2 + (Zs(i)-Zo)^2);
    B(i,1)=(Xs(i)-Xo)/D;
    B(i,2)=(Ys(i)-Yo)/D;
    B(i,3)=(Zs(i)-Zo)/D;
    idx=3+epoch(i);
    B(i,idx)=-c;
    F=pr(i) - D - c*rdt(epoch(i));
    f(i)=-F;
    end
  % B
  % f
  % W
  %condJ=cond(B)
  N=B'*W*B;
  %condN=cond(N)
  t=B'*W*f;
  Ni=inv(N);
  del=Ni*t;
  Qdd=Ni;
  Xo=Xo + del(1);
  Yo=Yo + del(2);
  Zo=Zo + del(3);
  rdt(1)=rdt(1) + del(4);
  rdt(2)=rdt(2) + del(5);
  rdt(3)=rdt(3) + del(6);
  rdt(4)=rdt(4) + del(7);

  disp_del=[iter; del(1); del(2); del(3)]
  disp_del2=[del(4) del(5) del(6) del(7)]
  v=f-B*del;
  phi=v'*W*v;
  if(abs(phi-old_phi)/phi < threshold)
    disp('we have converged');
    converged=1;
    break;
    end

  old_phi=phi;
  end

if(converged == 0)
  disp('we did not converge');
else
  % we converged
  % show results
  disp('receiver location (km)');
  [Xo Yo Zo]
  disp('receiver clock biases (usec)');
  [rdt']
  end
```

```
disp('residuals (km)');
v
rms=sqrt(v'*v/nobs)

% make global test and post adjustment statistics

r
test_stat=v'*W*v/(sigma0^2)
cv1=icdf('chi2',0.025,r)
cv2=icdf('chi2',0.975,r)
if((test_stat > cv1) && (test_stat < cv2))
  Sdd=sigma0^2 * Qdd;
  pass=1;
  disp('pass global test');
else
  sigma0_hat_sqr=v'*W*v/r;
  Sdd=sigma0_hat_sqr * Qdd;
  pass=0;
  disp('fail global test');
  end

% ok transform cofactor matrix to ENU from XYZ
% first get lat, lon, h from XYZ by iteration
% remember Xo,Yo,Zo are km

disp('convert XYZ to phi,lam,h');
lam=atan2(Yo,Xo);
a=6378137.0/1000;
f=1/298.257223563;
esqr=2*f-f^2;
e=sqrt(esqr);
phioo=atan(Zo/((1-e^2)*sqrt(Xo^2+Yo^2)))
keep_going=1;
while(keep_going == 1)
  N=a/sqrt(1-esqr*(sin(phioo))^2)
  new_phi=atan((Zo/sqrt(Xo^2 + Yo^2))*(1 + esqr*N*sin(phioo)/Zo))
  if(abs(new_phi - phioo) < 1.0e-06);
    keep_going=0;
    end
  phioo=new_phi
  end
N=a/sqrt(1-esqr*(sin(phioo))^2);
phi=phioo;
h=sqrt(Xo^2 + Yo^2)/cos(phi) - N;
disp('phi');
result=raddms(phi)
disp('lambda');
result=raddms(lam)
disp('h');
h

% rotate XYZ into ENU
% extract the submatrix for XYZ to transform
M=m1(pi/2 - phi)*m3(lam + pi/2);
J=M;
J
Sdd_enu=J*Sdd(1:3,1:3)*J';
mx=Sdd_enu(1:2,1:2);
sigz_sqr=Sdd_enu(3,3);
[V,D]=eig(mx);
if(D(1,1) < D(2,2))
  tmp=D(1,1);
  D(1,1)=D(2,2);
  D(2,2)=tmp;
  tmp=V(:,1);
  V(:,1)=V(:,2);
  V(:,2)=tmp;
  end

% 90%
P=0.90;
alpha=1-P;
if(pass == 1)
  alen=sqrt(D(1,1)*icdf('chi2',P,2))
  blen=sqrt(D(2,2)*icdf('chi2',P,2))
  theta=atan2(V(2,1),V(1,1))
  plot_ell(0,0,alen,blen,theta);
  title('90% confidence ellipse for e,n - pass global test');
  zfact=icdf('norm',1-alpha/2,0,1)
  half_width=zfact*sqrt(sigz_sqr)
  width=2*half_width
else
  alen=sqrt(D(1,1)*2*icdf('f',P,2,r))
```

```
  blen=sqrt(D(2,2)*2*icdf('f',P,2,r))
  theta=atan2(V(2,1),V(1,1))
  plot_ell(0,0,alen,blen,theta);
  title('90% confidence ellipse for e,n - fail global test');
  zfact=icdf('t',1-alpha/2,r)
  half_width=zfact*sqrt(sigz_sqr)
  width=2*half_width
  end
axis equal
```