anheader.m

```
function [Obs_types, ant_delta,ifound_types,eof] = anheader(file)
%ANHEADER Analyzes the header of a RINEX file and outputs
%        the list of observation types and antenna offset.
%        End of file is flagged 1, else 0. Likewise for the types.
%        Typical call: anheader('pta.96o')

%Kai Borre 09-12-96
%Copyright (c) by Kai Borre
%$Revision: 1.0 $   $Date: 1997/09/23  $

fid = fopen(file,'rt');
eof = 0;
ifound_types = 0;
Obs_types = [];
ant_delta = [];
```

```
while 1    % Gobbling the header
   line = fgetl(fid);
   answer = findstr(line,'END OF HEADER');
   if  ~isempty(answer), break; end;
   if (line == -1), eof = 1; break; end;
   answer = findstr(line,'ANTENNA: DELTA H/E/N');
   if ~isempty(answer)
      for k = 1:3
         [delta, line] = strtok(line);
         del = str2num(delta);
         ant_delta = [ant_delta del];
      end;
   end
   answer = findstr(line,'# / TYPES OF OBSERV');
   if ~isempty(answer)
      [NObs, line] = strtok(line);
      NoObs = str2num(NObs);
      for k = 1:NoObs
         [ot, line] = strtok(line);
         Obs_types = [Obs_types ot];
      end;
      ifound_types = 1;
   end;
end;
```

grabdata.m

```
function Obs = grabdata(fid, NoSv, NoObs)
%GRABDATA Positioned in a RINEX file at a selected epoch
%       reads observations of NoSv satellites

%Kai Borre 09-13-96
%Copyright (c) by Kai Borre
%$Revision: 1.0 $  $Date: 1997/09/23  $

global lin

Obs = zeros(NoSv, NoObs);

if NoObs <= 5   % This will typical be Turbo SII data
   for u = 1:NoSv
      lin = fgetl(fid);
      for k = 1:NoObs
         Obs(u,k) = str2num(lin(2+16*(k-1):16*k-2));
      end
   end
```

```
else          % This will typical be Z12 data
   Obs = Obs(:,[1 2 3 4 5]); % We cancel the last two columns 6 and 7
   NoObs = 5;
   for u = 1:NoSv
      lin = fgetl(fid);
      lin_doppler = fgetl(fid);
      for k = 1:NoObs  %%-1
         if isempty(str2num(lin(1+16*(k-1):16*k-2))) == 1, Obs(u,k) = nan;
        else                                %
        Obs(u,k) = str2num(lin(1+16*(k-1):16*k-2));
      end
     % Obs(u,NoObs) = str2num(lin(65:78));
   end
end
end
```

recpo_ls.m

```
function [pos, El, GDOP, basic_obs] = recpo_ls(obs,sats,time,Eph)
% RECPO_LS Computation of receiver position from pseudoranges
%          using ordinary least-squares principle

%Kai Borre 31-10-2001
%Copyright (c) by Kai Borre
%$Revision: 1.1 $  $Date: 2002/07/10  $

v_light = 299792458;
dtr = pi/180;
m = size(obs,1);  % number of svs
el = zeros(m,1);
% identify ephemerides columns in Eph
for t = 1:m
    col_Eph(t) = find_eph(Eph,sats(t),time);
end
% preliminary guess for receiver position and receiver clock offset
pos = zeros(4,1);
no_iterations = 6;
ps_corr = [];
sat_pos = [];
```

```
for iter = 1:no_iterations
    A = [];
    omc = []; % observed minus computed observation
    for i = 1:m
        k = col_Eph(i);
        tx_RAW = time - obs(i)/v_light;
        t0c = Eph(21,k);
        dt = check_t(tx_RAW-t0c);
        tcorr = (Eph(2,k)*dt + Eph(20,k))*dt + Eph(19,k);
        tx_GPS = tx_RAW-tcorr;
        dt = check_t(tx_GPS-t0c);
        tcorr = (Eph(2,k)*dt + Eph(20,k))*dt + Eph(19,k);
        tx_GPS = tx_RAW-tcorr;
        X = satpos(tx_GPS, Eph(:,k));
        if iter == 1
            traveltime = 0.072;
            Rot_X = X;
            trop = 0;
        else
            rho2 = (X(1)-pos(1))^2+(X(2)-pos(2))^2+(X(3)-pos(3))^2;
            traveltime = sqrt(rho2)/v_light;
            Rot_X = e_r_corr(traveltime,X);
            rho2 = (Rot_X(1)-pos(1))^2+(Rot_X(2)-pos(2))^2+(Rot_X(3)-pos(3))^2;
            [az,el,dist] = topocent(pos(1:3,:),Rot_X-pos(1:3,:));
            if iter == no_iterations, El(i) = el; end
            trop = tropo(sin(el*dtr),0.0,1013.0,293.0,50.0,...
                0.0,0.0,0.0);
        end
```

Solution to EASY3

```
        % subtraction of pos(4) corrects for receiver clock offset and
        % v_light*tcorr is the satellite clock offset
        if iter == no_iterations
            ps_corr = [ps_corr; obs(i)+v_light*tcorr-trop];
            sat_pos = [sat_pos; X'];
        end
        omc = [omc; obs(i)-norm(Rot_X-pos(1:3),'fro')-pos(4)+v_light*tcorr-trop];
        A = [A; (-(Rot_X(1)-pos(1)))/obs(i)...
                (-(Rot_X(2)-pos(2)))/obs(i) ...
                (-(Rot_X(3)-pos(3)))/obs(i) 1];
    end % i
    x = A\omc;
    pos = pos+x;
    if iter == no_iterations, GDOP = sqrt(trace(inv(A'*A)));
        %% two lines that solve an exercise on computing tdop
        % invm = inv(A'*A);
        % tdop = sqrt(invm(4,4))
    end
end % iter
basic_obs = [sat_pos ps_corr];
```

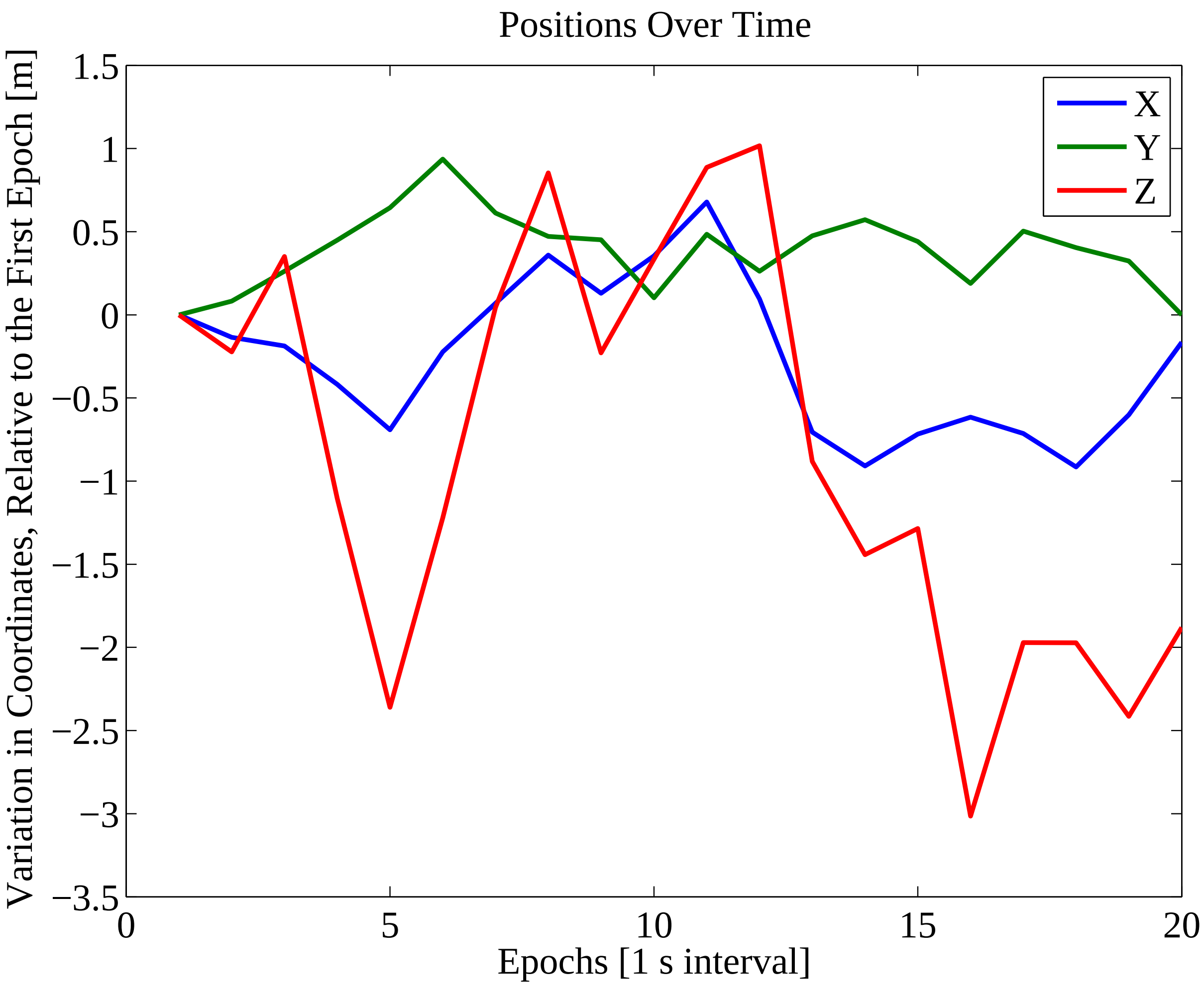
easy3.m

```
% EASY3   Read RINEX navigation file reformat into Matlab Eph matrix.
%         Open a RINEX observation file analyse the header and identify
%         observation types. The function fepoch_0 finds epoch time
%         and observed PRNs in an OK epoch (digit 0, RTK observations
%         will have a 2 in this place). Next we read the observations
%         and use recpo_ls to get a least-squares estimate for the
%         (stand alone) receiver position.

%Kai Borre 31-10-2001
%Copyright (c) by Kai Borre
%$Revision: 1.0 $  $Date: 2001/10/31  $

% Read RINEX ephemerides file and convert to
% internal Matlab format
rinexe('SITE247J.01N','eph.dat');
Eph = get_eph('eph.dat');

% We identify the observation file and open it
ofile1 = 'SITE247J.01O';
fid1 = fopen(ofile1,'rt');
```

```
[Obs_types1, ant_delta1, ifound_types1, eof11] = anheader(ofile1);
NoObs_types1 = size(Obs_types1,2)/2;
Pos = [];
% There are 20 epochs of data in ofile1
for q = 1:20
    [time1, dt1, sats1, eof1] = fepoch_0(fid1);
    NoSv1 = size(sats1,1);
    % We pick the observed P2 pseudoranges
    obs1 = grabdata(fid1, NoSv1, NoObs_types1);
    i = fobs_typ(Obs_types1,'P2');
    pos = recpo_ls(obs1(:,i),sats1,time1,Eph);
    Pos = [Pos pos];
end
me = mean(Pos,2);
fprintf('\nMean Position as Computed From 20 Epochs:')
fprintf('\n\nX: %12.3f  Y: %12.3f  Z: %12.3f', me(1,1), me(2,1), me(3,1))
plot((Pos(1:3,:)-Pos(1:3,1)*ones(1,q))','linewidth',2)
title('Positions Over Time','fontsize',16)
legend('X','Y','Z')
xlabel('Epochs [1 s interval]','fontsize',16)
ylabel('Variation in Coordinates, Relative to the First Epoch [m]','fontsize',16)
set(gca,'fontsize',16)
legend

print -depsc2 easy3
```

```
>> easy3
head_lines =
          8.00
noeph =
          7.00
status =
             0
ans =
             0

Mean Position as Computed From 20 Epochs:

X:  3427823.969  Y:   603665.739  Z:  5326881.602>>
```

Solution to EASY3



Solution to EASY3