Data Adj 1 HW #5

Generic 2D network adjustment assigned 9-Nov-09, due 18-nov-09

Make a matlab program to adjust a generic 2D network using distance and angle observations with given data file format — Sample data + results in data1_09_hw5.zip for debug. I email the "real" data to you on Tuesday (17-nov) & you run your code on it.

data specs:

obs.dat (text file)

seq #	obstype	i	j	k	obs1	obs2	obs3	σ
1	1	i_1	j_1	k_1	$O1_1^{(d)}$	$O2_1^{(0)}$	$O3_3^{(0)}$	σ_d
2	2	i_2	j_2	k_2	$O1_2^{(D)}$	$O2_2^{(M)}$	$O3_2^{(S)}$	σ_a (sec)
⋮	⋮	⋮	⋮	⋮				
n	1	i_n	j_n	k_n	$O1_n^{(d)}$	$O2_n^{(0)}$	$O3_n^{(0)}$	σ_d

↑ 1 = distance, 2 = angle

pnt.dat

point #	X	Y
1	X_1	Y_1
2	X_2	Y_2
3	X_3	Y_3
⋮	⋮	⋮
m	X_m	Y_m

control.dat

point #	coord component fixed	
1	1	(X_1)
1	2	(Y_1)
2	1	(X_2)
2	2	(Y_2)
⋮	⋮	⋮

- files have only numeric entries — read by "load" and parcel out to variable names
- point seq # and point ID are same — not the best but easy

helpful code hints :

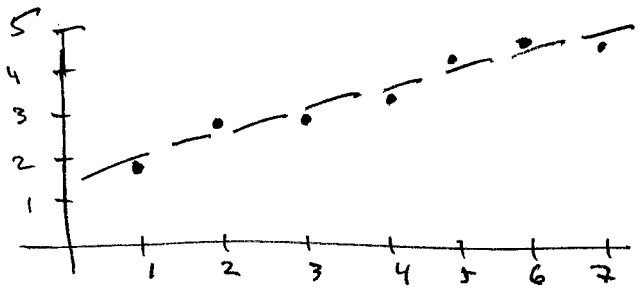
```
load obs.dat
load pnt.dat
load control.dat
[n, nc] = size(obs);
[npoint, nc] = size(pnt);
[ncon, nc] = size(control);
code = obs(:,2);
 i = obs(:,3);
 j = obs(:,4);
 k = obs(:,5);
 o1 = obs(:,6);
  :
 etc.
x = pnt(:,2);
y = pnt(:,3);
conpnt = control(:,1);
coordcomp = control(:,2);
m = npoint*2 - ncon;
r = n-m;
m0 = npoint*2;
W = zeros(n,n);
keep_going = 1;
```

```
while ( keep_going == 1 )
    B = zeros(n, nφ);          % parameter order !
    f = zeros(n, 1);
    for ic = 1:n
        if (code(ic) == 1)      % distance
            res = dist2d( o1(ic), i(ic), j(ic), X, Y );
            W(ic,ic) = 1 / sig(ic)^2 ;
            i_index = (i(ic) - 1) * 2 + 1
            j_index = (j(ic) - 1) * 2 + 1
            B(ic, i_index) = res(2) ;        % ∂F/∂xi
            B(ic, i_index+1) = res(3) ;      % ∂F/∂yi
            B(ic, j_index) = res(4) ;        % ∂F/∂xj
                  ⋮
            f(ic) = -res(1) ;
        end
        if (code(ic) == 2)      % angle
            % o1(ic), o2(ic), o3(ic) → a      DMS → rad
            % sig(ic) → sr                    sec → rad
            res = angle2d( a, i(ic), j(ic), k(ic), X, Y ) ;
            W(ic,ic) = 1 / (sr^2) ;
            i_index = (i(ic)-1) * 2 + 1
            j_index = ...
            k_index = ...
            B(ic, i_index) = res(2) ;        % ∂F/∂xi
                  ⋮
            f(ic) = - res(1) ;
        end
    end
```

```
% purge columns of B
B = elim_col ( B, purge)   % purge = array of col#'s to purge
% I/O LS
 del = (B'WB)^-1 B'Wf ...,
 v =  f - B * del ...,
% inflate delta vector
 del = insert_zerV (del, purge);
 for ii = 1: npoint
     index = (ii-1) * 2 + 1 ;
     X(ii) = X(ii) + del (index);
     Y(ii) = Y(ii) + del (index+1);
     end
%check v'Wv or del for convergence + set "keep-going"
end % while
```

- do global test, 2-sided @ $\alpha = .05$
- show results
- show V (angles in arc-sec)
- show adjusted point coords
- careful with variable names ⇐ "dual use"

GLS problem



X	Y
1.0	1.8
2.0	2.8
3.0	2.9
4.0	3.2
5.0	4.0
6.0	4.4
7.0	4.5

X & Y both observed

Solve GLS adjustment problem