```
A=reshape([1,0,0,-1,1,-1,0,0,1,0,0,0,0,1,-1,0,0,1,0,0,0,0,1,-1,0,0,1,0,0,0,0,1],4,8)'

b=[1.631,1.204,2.947,3.186,1.735,-2.778,-1.449,1.321]';

[u,s,v]=svd(A);

st=s';
for i=1:4
st(i,i)=1/st(i,i);
end;

sol2=v*st*u'*(b);

% look at errors:

[A*sol2-b]

std(A*sol2-b)


%%%%%%%%%%%%%%%%%%%%%%%%%% NOW another approach

P=A*((A'*A)\A')
IP=eye(8)-P;

% You prove I - P = I - U(:,1..m)*U'(:,1..m)

% Notice, idempotent:

IP^2-IP

% Now onward

[v,e]=eig(IP);

diag(e)

% notice  4 nonzero eigenvalues


% inf norm solution: pin e_{1,2,4,6} at:

ev=[-0.00475,.00475,0,.00475,0,.00475,0,0]'

rhs=-IP*(b+ev);

% remaining free variables - use LS (not required, could choose other)
% for a change, use the qr decomp (return to later, see matlab qr).

IPs=IP(:,[3,5,7,8]);

        r=qr(IPs);
        xs = r\(r'\(IPs'*rhs))
        res = rhs - IPs*xs;
        iter = r\(r'\(IPs'*res));
        xs = xs + iter

% Takes many iterations, converges to:

   0.00000308764268
   0.00325308764268
  -0.00349691235732
  -0.00024691235732

% vs.  IPs\rhs in matlab, which gives

   0.00025000000000
```

```
   0.00350000000000
  -0.00325000000000
                  0

%  sum of squares of first:   2.287195049146928e-05
%                 of second:  2.287500000000000e-05

ev(3)=   0.00000308764268;
ev(5)=   0.00325308764268;
ev(7)=  -0.00349691235732;
ev(8)=  -0.00024691235732;

% check projection now:

IP*(b+ev)

% Yes -- b+errors    now has no projection in null space of A

[u,s,v]=svd(A);

sol=v*st*u'*(b+ev);

[A*sol,b+ev]

% compare errors of the two data fits:

[ev,A*sol2-b]

% compare misfit in L2 basis:

std(ev)
std(A*sol2-b)
```