Waveform Fitting of Cross-Spectra to Determine Phase Velocity Using Aki's (1957) Formula

William Menke, February 6, 2013

Aki (1957) gives the following expression for the azimuthally-averaged, normalized cross-spectrum:

$$\rho(\omega) = A\,J_0\left(\frac{\omega r}{c(\omega)}\right) \quad \text{with} \quad A = 1$$

with angular frequency ω, inter-station distance r and phase velocity $c(\omega)$. We have added the amplitude factor A to allow for the possibility that an observed cross-spectrum may not be correctly normalized.

Ekstrom et al. (2009) demonstrate that this formula can be used to determined the phase velocity at the frequencies, say $\Omega_\mathrm{i}, i = 1 \cdots N$, at which the cross-spectrum is zero:

$$\text{when } \omega_i = \Omega_\mathrm{i} \;\; \text{then} \;\; \rho(\Omega_\mathrm{i}) = 0 \;\; \text{and} \;\; C_i \equiv c(\Omega_\mathrm{i}) = \frac{\Omega_\mathrm{i} r}{Z_{j+k}}$$

Here Z_j is the j-th zero of the Bessel function; that is, $J_0(Z_j) = 0$. Owing to the finite bandwidth of observations, the cross-spectrum $\rho(\omega)$ is typically measured only for a finite frequency band, and so only a subset of zero-crossings can be estimated. The integer k is introduced to correlate the lowest observed zero of the cross-spectrum with the corresponding zero of the Bessel function. While the value of k is not immediately known, Ekstrom et al. (2009) develop a trial-and-error procedure for determining it, based on prior bounds on the phase velocity; that is, an incorrect value for k leads to an implausible estimate of C_i. Once the phase velocity is known at a sequence of discrete frequencies, its value at other frequencies can be estimated by interpolation.

Aki's formula predicts the complete functional form of the cross-spectrum, not just the location of the zero crossings. We may ask then whether additional information about the phase velocity can be determined via "waveform fitting"; that is, determining the phase velocity $c(\omega)$ and amplitude A that minimizes the L_2 error,

$$E = \int_{\omega_{min}}^{\omega_{max}} \{\rho^{obs}(\omega) - \rho^{pre}(\omega)\}^2 \, d\omega$$

between the observed cross-spectrum $\rho^{obs}(\omega)$ and the one predicted by Aki's method, ρ^{pre}. Here, $(\omega_{min}, \omega_{max})$ is the frequency-band of the observations. Optional constraints can be imposed on the minimization, such as $c(\Omega_\mathrm{i}) \approx C_i$ (the zero crossing estimates are approximately satisfied), or $d^2c/d\omega^2 \approx 0$ (the phase velocity smoothly varies with frequency), or both. In addition to providing more information about the phase velocity, itself, such a minimization would also be able to provide formal estimates of the variance of the results, provided that an estimate of the variance σ_ρ^2 of the cross-spectrum is available.

We solve the minimization problem by first approximating the cross-spectrum and phase velocity as frequency series $\boldsymbol{\rho}$ and $\mathbf{c}$, respectively, say with Nvalues and with frequency spacing $\Delta\omega$. We then linearize the problem around an initial estimate of the unknowns $\left(\mathbf{c}^{(0)}, A^{(0)}\right)$. The data kernel $\mathbf{G}$ of the least squares formulation contains the derivatives **:**

$$G_{ij} = \left.\frac{d\rho_i}{dc_j}\right|_{\mathbf{c}^{(p)},A^{(p)}} = A^{(p)}\,\delta_{ij}\left(\frac{\omega_j r}{\left(c_j{}^{(p)}\right)^2}\right) J_1\left(\frac{\omega_j r}{c_j{}^{(p)}}\right) \quad \text{for} \quad (1 \le i \le N) \text{ and } (1 \le j \le N)$$

$$G_{ij} = A^{(p)}\,J_0\left(\frac{\omega_j r}{c_j{}^{(p)}}\right) \text{ for } \quad (1 \le i \le N) \text{ and } j = N+1$$

Optionally, we can add a constraint equation of the form $\mathbf{Hm} = \mathbf{h}$. In the example, below, we impose smoothness on the phase velocities by seting $\mathbf{H}$ to the discrete form of the second derivative operator:

$$\mathbf{h} = 0 \quad \text{and} \quad H_{ij} = \begin{cases} (\Delta\omega)^{-2} & i \le N \text{ and } j = i-1 \\ -2(\Delta\omega)^{-2} & i \le N \text{ and } j = i \\ (\Delta\omega)^{-2} & i \le N \text{ and } j = i+1 \\ 0 & \text{otherwise} \end{cases}$$

The iterative Generalize Least Squares solution is then obtained by iterating the formulas

$$\rho_i{}^{(p)} = A^{(p)}\,J_0\left(\frac{\omega_i r}{c_i{}^{(p)}}\right)$$

$$\Delta\boldsymbol{\rho}^{(p)} = \boldsymbol{\rho}^{obs} - \boldsymbol{\rho}^{(p)}$$

$$\mathbf{M}^{(\mathrm{p})} = \left[\mathbf{G}^{(\mathrm{p})\mathrm{T}}\mathbf{G}^{(\mathrm{p})} + \sigma_{\mathrm{h}}^{-2}\mathbf{H}^{(\mathrm{p})\mathrm{T}}\mathbf{H}^{(\mathrm{p})}\right]^{-1}\mathbf{G}^{(\mathrm{p})\mathrm{T}}$$

$$\begin{bmatrix} \mathbf{c}^{(\mathrm{p}+1)} \\ A^{(p+1)} \end{bmatrix} = \mathbf{M}^{(\mathrm{p})}\Delta\boldsymbol{\rho}^{(p)}$$

with $p = (0, 1, 2, \cdots q)$ until convergence is reached. Here σ_{h}^2 is the variance of the elements of $\mathbf{h}$; that is, the degree of certainly in the $d^2c/d\omega^2 = 0$ constraint. A small σ_{h}^2 leads to a smooth solution and a large σ_{h}^2 to a rough one. The covariance of the solution can be approximated as

$$\begin{bmatrix} \mathrm{cov}(\boldsymbol{\rho},\boldsymbol{\rho}) & \mathrm{cov}(\boldsymbol{\rho},\mathrm{A}) \\ \mathrm{cov}(\mathrm{A},\boldsymbol{\rho}) & \mathrm{cov}(\mathrm{A},\mathrm{A}) \end{bmatrix} = \sigma_\rho^2\mathbf{M}^{(\mathrm{p})}\mathbf{M}^{(\mathrm{p})\mathrm{T}}$$

and the resolution matrix as $\mathbf{R} = \mathbf{M}^{(\mathrm{p})}\mathbf{G}$.

References

Aki, K., 1957. Space and time spectra of stationary stochastic waves, with special reference to microtremors, Bull. Earthq. Res. Inst., 35, 415–457.

Göran Ekström, G., G.A. Abers and S.C. Webb, Determination of surface-wave phase velocities across USArray from noise and Aki's spectral formulation, Geophysical Research Letters, 2009. DOI: 10.1029/2009GL039131

Example

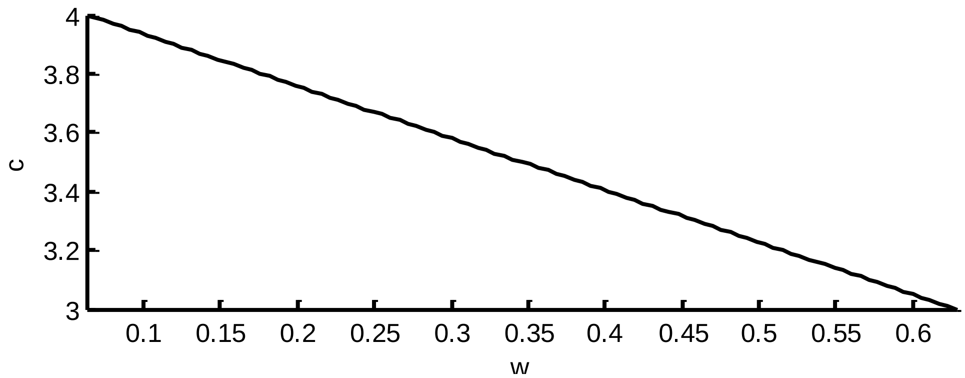

Figure 1. True (hypothetical) phase velocity, $c^{true}(\omega)$.

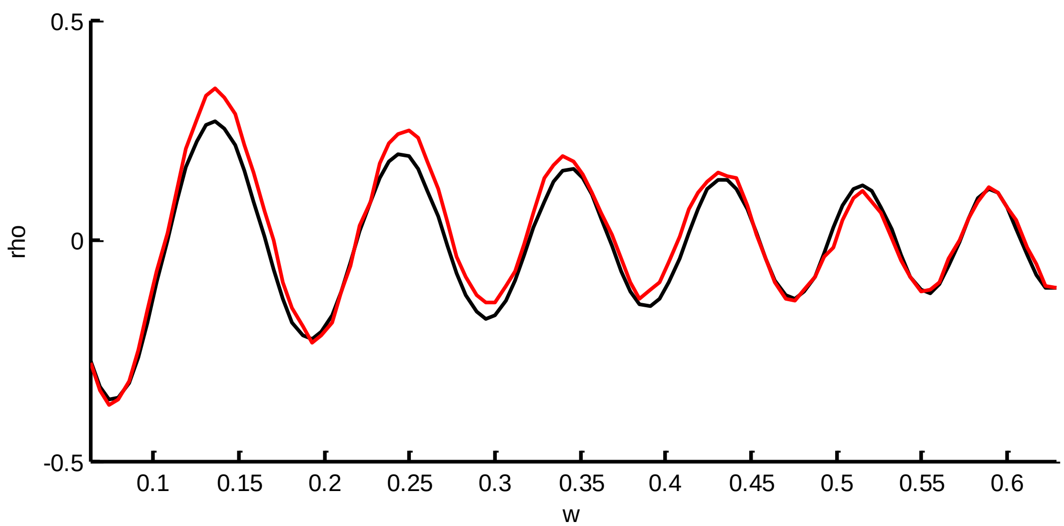

Figure 2. True cross-spectrum, $\rho^{true}(\omega)$ (black curve) and observed cross-spectrum $\rho^{obs}(\omega) = \rho^{true}(\omega) + n(\omega)$ (red curve), where $n(\omega)$ is band-limited noise.

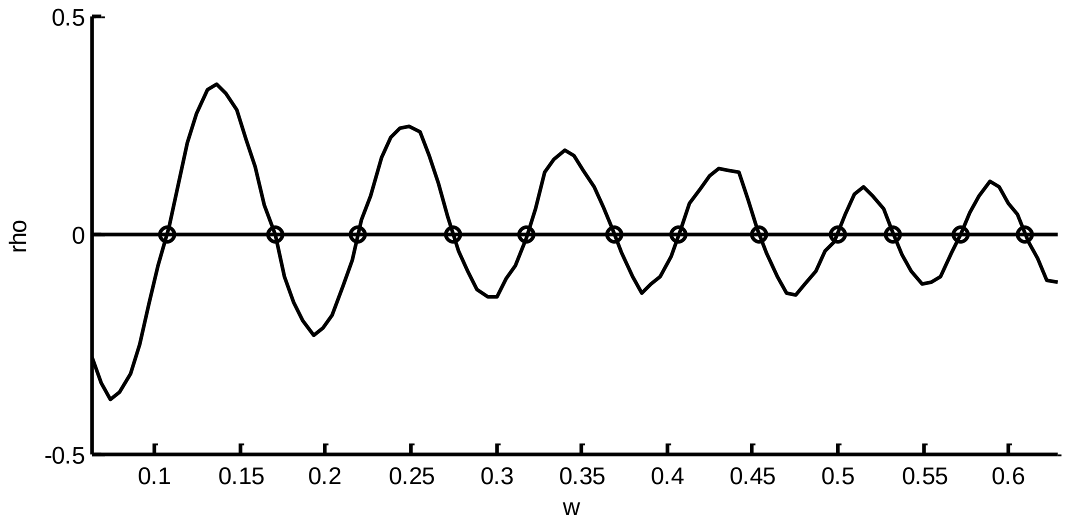

Figure 3. Zero-crossings (black circles) of the observed cross-spectrum $\rho^{obs}(\omega)$ (black curve).

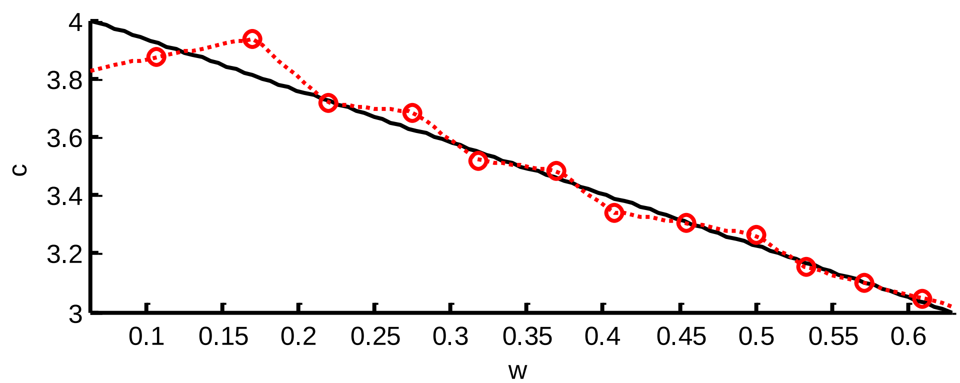

Figure 4. True phase velocity $c^{true}(\omega)$ (black curve), estimates C_i based on zero-crossings (red circles) and linear interpolation (dotted red curve).

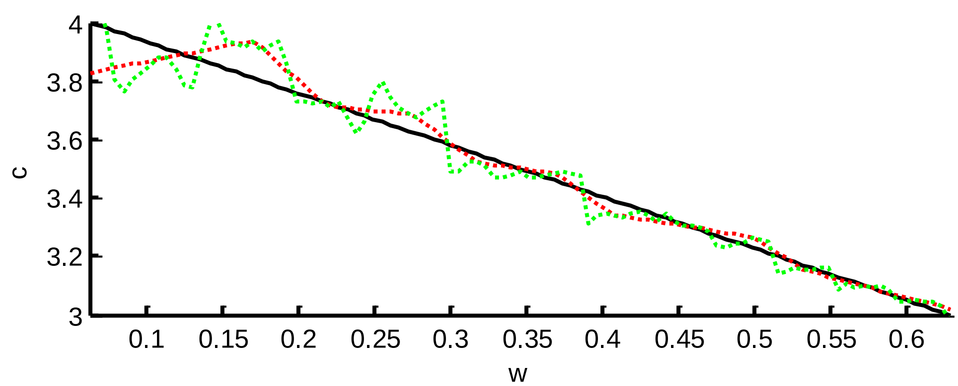

Figure 5. True phase velocity $c^{true}(\omega)$ (black curve), estimates based on interpolating zero-crossings (dotted red curve) and results of waveform fitting (green curve).

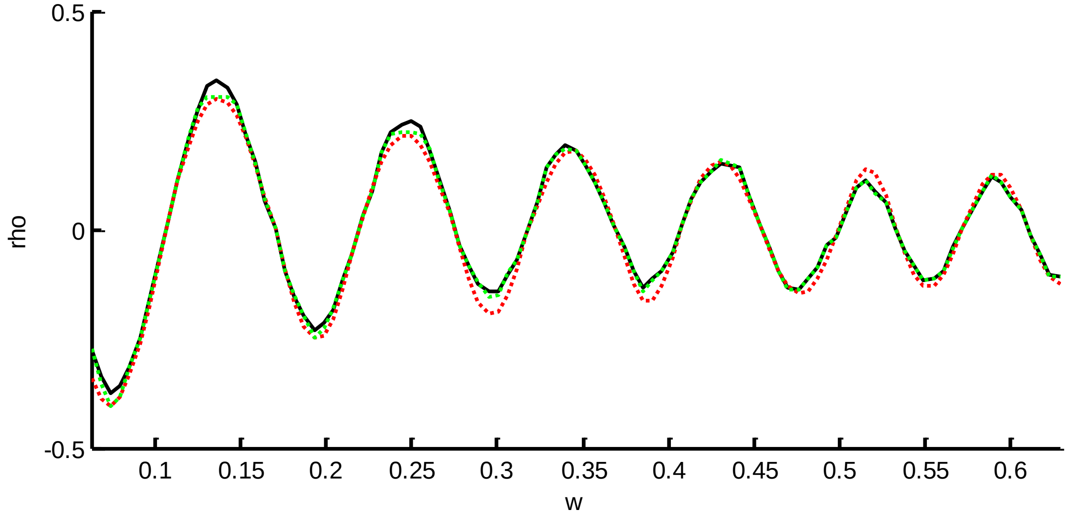

Figure 6. True cross-spectrum, $\rho^{true}(\omega)$ (black curve) and cross-spectrum predicted on the basis of zero-crossings (red curve) and waveform fitting (green curve). The variance of the fit has decreased by about 85% between the trial solution ($p = 0$) and the final solution ($p = 30$).

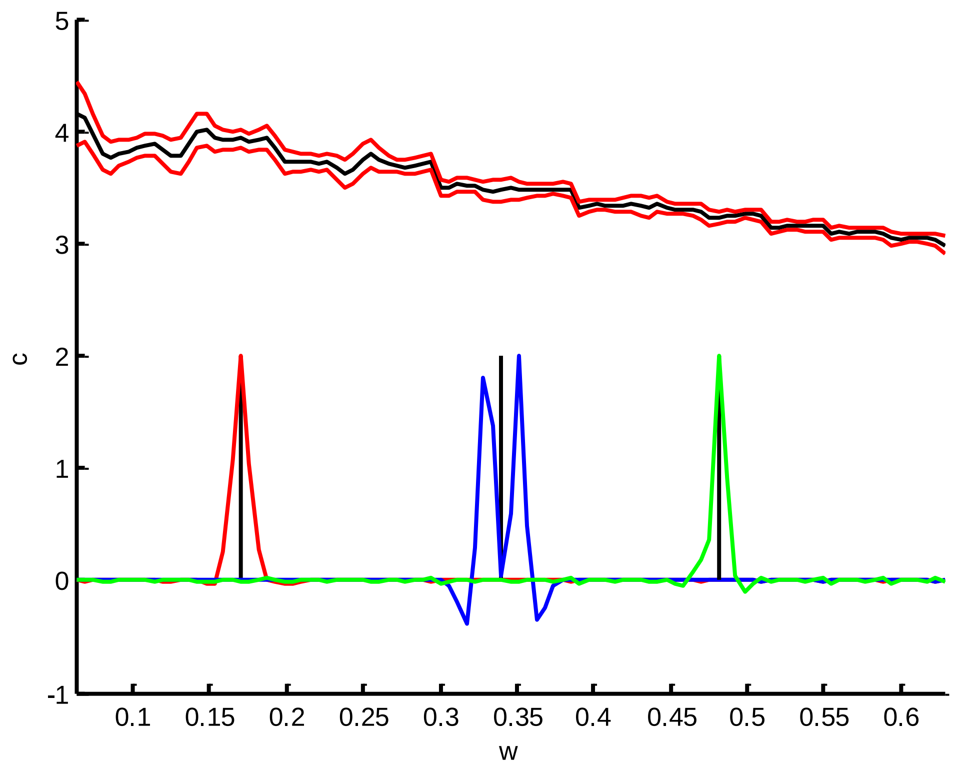

Figure 7. (Top) Phase velocity estimated by waveform fitting (black curve) with 99% confidence bounds (red curves). (Bottom) Resolving kernels (colored curves) for phase velocity at three frequencies (vertical bar). The bimodal shape of the central kernels is related to the presence of an extremum of the cross-spectrum at that frequency. The cross-spectrum contributes no information about the phase velocity when at frequencies where its slope is zero, at least in a linearized inversion, since small perturbations in phase velocity do not change its shape.

```
clear all;
global G H;

% set up frequencies
N=101;
Tmin = 10;
Tmax = 100;
wmin = 2*pi/Tmax;
wmax = 2*pi/Tmin;
w = wmin + (wmax-wmin)*[0:N-1]'/(N-1);
T = 1./w;

% true phase velocity
c1 = 4.0;
c2 = 3.0;
c = c1 + (c2-c1)*[0:N-1]'/(N-1);
A = 0.9;

% plot phase velocity
figure(1);
clf;
set(gca,'LineWidth',2);
hold on;
axis( [wmin, wmax, min(c), max(c)] );
plot( w, c, 'k-', 'LineWidth', 2 );
xlabel('w');
ylabel('c');

% true correlation function
rho_true = zeros(N,1);
r=200;
nu=0;
rho_true = A*besselj(nu,w*r./c);

% make correlated noise
s_rho=0.1;
uncorrelated_noise = random('Normal',0,s_rho,N,1);
L = 10;
correlated_noise = conv( uncorrelated_noise,
ones(L,1)/L );
correlated_noise = correlated_noise(1:N);
```

```
sc = std(correlated_noise);

% synthetic observed correlogram, equals true plus
noise
rho_obs = rho_true + correlated_noise;
figure(2);
clf;
set(gca,'LineWidth',2);
hold on;
axis( [wmin, wmax, -0.5, 0.5 ] );
plot( w, rho_true, 'k-', 'LineWidth', 2 );
plot( w, rho_obs, 'r-', 'LineWidth', 2 );
xlabel('w');
ylabel('rho');

% find zero crosings of observed correlation function
% part 1: bracket zero crossings
wL = w(2:N);
wR = w(1:N-1);
rhoL = rho_obs(2:N);
rhoR = rho_obs(1:N-1);
iz=find( (rhoL.*rhoR)<0 );
Nz = length(iz);
% part 2, linear interpolations
wz = (rhoR(iz).*wL(iz) - rhoL(iz).*wR(iz)) ./
(rhoR(iz)-rhoL(iz));

% plot brackets
figure(3);
clf;
set(gca,'LineWidth',2);
hold on;
axis( [wmin, wmax, -0.5, 0.5 ] );
plot( w, rho_obs, 'k-', 'LineWidth', 2 );
xlabel('w');
ylabel('rho');
plot( [wmin, wmax]', [0, 0]', 'k-', 'LineWidth', 2 );
plot( wL(iz), rhoL(iz), 'ro', 'LineWidth', 2 );
plot( wR(iz), rhoR(iz), 'ro', 'LineWidth', 2 );

% plot zero crossings
```

```
figure(4);
clf;
set(gca,'LineWidth',2);
hold on;
axis( [wmin, wmax, -0.5, 0.5 ] );
plot( w, rho_obs, 'k-', 'LineWidth', 2 );
xlabel('w');
ylabel('rho');
plot( wz, zeros(Nz,1), 'ko', 'LineWidth', 2 );
plot( [wmin, wmax]', [0, 0]', 'k-', 'LineWidth', 2 );

% zeros of Jo
Nzmax = 2*Nz;
bz = zeros(Nzmax,1);
bz = besselzero(nu,Nzmax,1);

% correlated zero crossings of correlation function and
Jo
% by looking for plausible phase velocities
cmin = 2.8;
cmax = 4.2;
for k=[0:5]
     cest = r*wz./bz(1+k:Nz+k);
     if( (min(cest)>=cmin) && (max(cest)<=cmax) )
         break
     end
end

% plot phase velocities associated with zero crossings
figure(5);
clf;
set(gca,'LineWidth',2);
hold on;
axis( [wmin, wmax, min(c), max(c)] );
plot( w, c, 'k-', 'LineWidth', 2 );
xlabel('w');
ylabel('c');
plot( wz, cest, 'ro', 'LineWidth', 2 );

% interpolate phase velocities to all frequencies
% predict correlation function
```

```
% compute error
c0 = interp1( wz, cest, w, 'linear', 'extrap' );
plot( w, c0, 'r:', 'LineWidth', 2 );
A0 = 1.1;
rho0 = A0*besselj(nu,w*r./c0);
Drho = rho_obs - rho0;
A0 = (max(rho_obs)-min(rho_obs))/((max(rho0)-
min(rho0))/A0);
E = Drho'*Drho;
E0 = E;

% rho = besselj(nu,w*r./c);
% y = wr c^-1
% d/dy Jo(y) = - J1(y)
% so d(rho)/dc = d(rho)/dy  dy/dc
% = -J1(wr/c) * -wr c^-2
% = J1(wr/c) wr/c^2

% starting values of linearized inversion
cp = c0;
Ap = A0;
rhop = rho0;

% matrix H of prior constraints
% first half, smallness
% second half, smoothness
Ntop = N+1;
Nbot = N-2;
H = spalloc(Ntop+Nbot,N+1,5*N);
epsi1=0.01;
k=1;
for i=[1:N+1]
    H(k,i) =  epsi1;
    k=k+1;
end
epsi2 = 0.5;
for i=[2:N-1]
    H(k,i-1) =  epsi2;
    H(k,i) =  -2*epsi2;
    H(k,i+1) =  epsi2;
    k=k+1;
```

```
end

h = zeros(Ntop+Nbot,1);

% linearized inversion
% patch Feb 5, 2014.  Invert for overall amplitude, A
% adds one model parameter (make it the last model
parameter
% d(rho)/dc is left part of data kernel
% d(rho)/dA is right-most column
for iter = [1:120]
    mydiag = Ap * besselj(1,w*r./cp) .* w*r .* (cp.^(-
2));
    G = spalloc( N, N+1, 3*N );
    for i=[1:N]
        G(i,i) =  mydiag(i);
    end
    for i=[1:N]
        G(i,N+1) =  besselj(nu,w(i)*r/c0(i));
    end
    Drho = rho_obs - rhop;
    E = Drho'*Drho;
    Dm = bicg( @weightedleastsquaresfcn2, G'*Drho+H'*h,
1e-5, 4*N );
    cp = cp + Dm(1:N);
    Ap = Ap + Dm(N+1);
    myarg = w*r./cp;
    rhop = Ap*besselj(nu,myarg);
end
Drho = rho_obs - rhop;
E = Drho'*Drho;
fprintf('Variance reduction %f\n', 1-(E/E0) );

% plot starting and estimated phase velocity
figure(6);
clf;
set(gca,'LineWidth',2);
hold on;
axis( [wmin, wmax, min(c), max(c)] );
plot( w, c, 'k-', 'LineWidth', 2 );
xlabel('w');
```

```
ylabel('c');
plot( w, c0, 'r:', 'LineWidth', 2 );
plot( w, cp, 'g:', 'LineWidth', 2 );

% plot observed, starting and estimated corellation
function
figure(7);
clf;
set(gca,'LineWidth',2);
hold on;
axis( [wmin, wmax, -0.5, 0.5 ] );
plot( w, rho_obs, 'k-', 'LineWidth', 2 );
plot( w, rho0, 'r:', 'LineWidth', 2 );
plot( w, rhop, 'g:', 'LineWidth', 2 );
xlabel('w');
ylabel('rho');

% resolution matrix
Gmg = (G'*G + H'*H)\G';
R = Gmg*G;

% plot selected rows of resolution matrix
figure(8);
clf;
set(gca,'LineWidth',2);
hold on;
axis( [wmin, wmax, -1, 5] );
xlabel('w');
ylabel('c');
plot( w, cp, 'k-', 'LineWidth', 2 );

k=floor(N/5);
plot( [w(k), w(k)]', [0,2]', 'k-', 'LineWidth', 2 );
maxR = max(R(k,1:N));
plot( w, 2*R(k,1:N)'/maxR, 'r-', 'LineWidth', 2 );

k=floor(N/2);
plot( [w(k), w(k)]', [0,2]', 'k-', 'LineWidth', 2 );
maxR = max(R(k,1:N));
plot( w, 2*R(k,1:N)'/maxR, 'b-', 'LineWidth', 2 );
```

```
k=floor(3*N/4);
plot( [w(k), w(k)]', [0,2]', 'k-', 'LineWidth', 2 );
maxR = max(R(k,1:N));
plot( w, 2*R(k,1:N)'/maxR, 'g-', 'LineWidth', 2 );

% covariance matrix
cov_m = (sc^2) * Gmg * (Gmg');
cov_c = cov_m(1:N,1:N);
sigma2_c = diag(cov_c);
sigma_c = sqrt(sigma2_c);

% plot error bars for phase velocity
plot( w, cp+3*sigma_c, 'r-', 'LineWidth', 2 );
plot( w, cp-3*sigma_c, 'r-', 'LineWidth', 2 );

k = find( diff(myarg)<0 );
Nk = length(k);
fprintf('%d points are backward\n', Nk );
fprintf('Amplituded true %f starting %f final %f\n', A,
A0, Ap );



function y = weightedleastsquaresfcn2(v,transp_flag)
global G H;
temp1 = G*v;
temp2 = H*v;
y = G'*temp1 + H'*temp2;
return
```