```
function [a_uz,a_Sxz,a_Syz] = easy_anti_coeff(G,xi,yi,xj,yj,aj)
%  Calculates displacement and stress influence coefficients for
%  the anti-plane strain displacement discontinuity script
%  easy_anti.
%  Displacement discontinuities have a unit magnitude slip = b.
%  For anti-plane strain elements parallel to x-axis with stress
%  boundary conditions
%  Reference frame: x = horizontal, y = vertical, z = "out of
%  page"
%  A(i,j) = influence on observation pt i due to an effect at
%  element j
%  G = shear modulus
%  xi = x-coordinate of point feeling the influence
%  yi = x-coordinate of point feeling the influence
%  xj = x-coordinate of midpoint of element causing the
influence
%  yj = y-coordinate of midpoint of element causing the
influence
%  Last revised on 2/18/03

b = 1;
b_2pi     = b/(2*pi);
Gb_2pi    = G*b/(2*pi);

% Set up intial matrices used to find influence coefficients
% [Aij] [dj] = [bi],          [mxn]*[nx1] = [mx1]
% The influence coefficient matrices are mxn matrices, where
% m (the number of rows) is the # of observation points
% n (the number of columns) is the # of boundary elements
% The number of columns in A is the number of rows in [d]
% The number of rows in A is the number of rows in [b]
% Distances are from the jth element to the ith element
m    = length(xi);            % m = # of observation points
n    = length(xj);            % n = # of boundary elements
% All the matrices below have dimensions of mxn
xi   = repmat(xi(:),1,n);     % Row i = xi (i = 1:m)
yi   = repmat(yi(:),1,n);     % Row i = xi (i = 1:m)
xj   = repmat((xj(:))',m,1);  % Column j = xj (j = 1:n)
yj   = repmat((yj(:))',m,1);  % Column j = xj (j = 1:n)
aj   = repmat((aj(:))',m,1);  % Column j = xj (j = 1:n)

% Contribution from "positive" end ("p")
xp        = xi - (xj+aj);
yp        = yi - yj;
rp2  = xp.*xp + yp.*yp;
Syzp      =  Gb_2pi.*(xp./rp2);
Sxzp      = -Gb_2pi.*(yp./rp2);
uzp  = b_2pi*atan2(yp,xp);
```

```
% Contribution from "negative" end ("n")
xn        = xi - (xj-aj);
yn        = yi - yj;
rn2       = xn.*xn + yn.*yn;
Syzn =  Gb_2pi*(xn./rn2);
Sxzn      = -Gb_2pi*(yn./rn2);
uzn  = b_2pi*atan2(yn,xn);

% Total contribution (note the minus sign)
a_Syz     = Syzp - Syzn;
a_Sxz     = Sxzp - Sxzn;
a_uz      = uzp - uzn;
```