

```
disp(' ')
disp(' srs.m ver 2.0 July 3, 2006')
disp(' by Tom Irvine Email: tomirvine@aol.com')
disp(' ')
disp(' This program calculates the shock response spectrum')
disp(' of an acceleration time history, which is pre-loaded into Matlab.')
disp(' The time history must have two columns: time(sec) & acceleration')
disp(' ')
%
clear t;
clear y;
clear yy;
clear n;
clear fn;
clear a1;
clear a2;
clear b1;
clear b2;
clear jnum;
clear THM;
clear resp;
clear x_pos;
clear x_neg;
%
iunit=input(' Enter acceleration unit: 1= G 2= m/sec^2 ');
%
disp(' ')
disp(' Select file input method ');
disp(' 1=external ASCII file ');
disp(' 2=file preloaded into Matlab ');
file_choice = input('');
%
if(file_choice==1)
    [filename, pathname] = uigetfile('*.');
    filename = fullfile(pathname, filename);
%
    fid = fopen(filename,'r');
    THM = fscanf(fid,'%g %g',[2 inf]);
    THM=THM';
else
    THM = input(' Enter the matrix name: ');
end
%
t=double(THM(:,1));
y=double(THM(:,2));
%
tmx=max(t);
tmi=min(t);
n = length(y);
%
out1 = sprintf('\n %d samples \n',n);
disp(out1)
%
dt=(tmx-tmi)/(n-1);
sr=1./dt;
```

```
%
out1 = sprintf(' SR = %g samples/sec    dt = %g sec \n',sr,dt);
disp(out1)
%
fn(1)=input(' Enter the starting frequency (Hz) ');
if fn(1)>sr/30.
    fn(1)=sr/30.;
end
%
idamp=input(' Enter damping format: 1= damping ratio  2= Q ');
%
disp(' ')
if(idamp==1)
    damp=input(' Enter damping ratio (typically 0.05) ');
else
    Q=input(' Enter the amplification factor (typically Q=10) ');
    damp=1./(2.*Q);
end
%
disp(' ')
disp(' Select algorithm: ')
disp(' 1=Kelly-Richman  2=Smallwood ');
ialgorithm=input(' ');
%
tmax=(tmx-tmi) + 1./fn(1);
limit = round( tmax/dt );
n=limit;
yy=zeros(1,limit);
for i=1:length(y)
    yy(i)=y(i);
end
%
disp(' ')
disp(' Calculating response..... ')
%
% SRS engine
%
for j=1:1000
%
    omega=2.*pi*fn(j);
    omegad=omega*sqrt(1.-(damp^2));
    cosd=cos(omegad*dt);
    sind=sin(omegad*dt);
    domegadt=damp*omega*dt;
%
    if(ialgorithm==1)
        a1(j)=2.*exp(-domegadt)*cosd;
        a2(j)=-exp(-2.*domegadt);
        b1(j)=2.*domegadt;
        b2(j)=omega*dt*exp(-domegadt);
        b2(j)=b2(j)*( (omega/omegad)*(1.-2.*(damp^2))*sind -2.*damp*cosd );
        b3(j)=0;
%
    else
        E=exp(-damp*omega*dt);
```

```

        K=omegad*dt;
        C=E*cos(K);
        S=E*sin(K);
        Sp=S/K;
%
        a1(j)=2*C;
        a2(j)=-E^2;
        b1(j)=1.-Sp;
        b2(j)=2.*(Sp-C);
        b3(j)=E^2-Sp;
    end
    forward=[ b1(j), b2(j), b3(j) ];
    back    =[      1, -a1(j), -a2(j) ];
%
%
    resp=filter(forward,back,yy);
%
%
    x_pos(j)= max(resp);
    x_neg(j)= min(resp);
%
%
    jnum=j;
    if fn(j) > sr/8.
        break
    end
    fn(j+1)=fn(1)*(2. ^ (j*(1./12.)));
end
%
% Output options
%
disp(' ')
disp(' Select output option ');
choice=input(' 1=plot only   2=plot & output text file ');
disp(' ')
%
if choice == 2
%%
    [writefname, writepname] = uiputfile('*', 'Save SRS data as');
    writefname = fullfile(writepname, writefname);
    writedata = [fn' x_pos' (abs(x_neg))' ];
    fid = fopen(writefname, 'w');
    fprintf(fid, ' %g %g %g\n', writedata);
    fclose(fid);
%%
%   disp(' Enter output filename ');
%   SRS_filename = input(' ', 's');
%
%   fid = fopen(SRS_filename, 'w');
%   for j=1:jnum
%       fprintf(fid, '%7.2f %10.3f %10.3f \n', fn(j), x_pos(j), abs(x_neg(j)));
%   end
%   fclose(fid);
end
%
% Plot SRS
%
disp(' ')

```

```
disp(' Plotting output..... ')
%
% Find limits for plot
%
srs_max = max(x_pos);
if max( abs(x_neg) ) > srs_max
    srs_max = max( abs(x_neg) );
end
srs_min = min(x_pos);
if min( abs(x_neg) ) < srs_min
    srs_min = min( abs(x_neg) );
end
%
figure(1);
plot(fn,x_pos,fn,abs(x_neg),'-.');
%
if iunit==1
    ylabel('Peak Accel (G)');
else
    ylabel('Peak Accel (m/sec^2)');
end
xlabel('Natural Frequency (Hz)');
Q=1./(2.*damp);
out5 = sprintf(' Acceleration Shock Response Spectrum Q=%g ',Q);
title(out5);
grid;
set(gca,'MinorGridLineStyle','none','GridLineStyle',':', 'XScale','log','YScale','log');
legend ('positive','negative',2);
%
ymax= 10^(round(log10(srs_max)+0.8));
ymin= 10^(round(log10(srs_min)-0.6));
%
fmax=max(fn);
fmin=fmax/10.;
%
fmax= 10^(round(log10(fmax)+0.5));
%
if fn(1) >= 0.1
    fmin=0.1;
end
if fn(1) >= 1
    fmin=1;
end
if fn(1) >= 10
    fmin=10;
end
if fn(1) >= 100
    fmin=100;
end
axis([fmin,fmax,ymin,ymax]);
%
disp(' ')
disp(' Plot pseudo velocity? ');
vchoice=input(' 1=yes 2=no ');
if(vchoice==1)
```

```
figure(2);
%
% Convert to pseudo velocity
%
for j=1:jnum
    if iunit==1
        x_pos(j)=386.*x_pos(j)/(2.*pi*fn(j));
        x_neg(j)=386.*x_neg(j)/(2.*pi*fn(j));
    else
        x_pos(j)=x_pos(j)/(2.*pi*fn(j));
        x_neg(j)=x_neg(j)/(2.*pi*fn(j));
    end
end
%
srs_max = max(x_pos);
if max( abs(x_neg) ) > srs_max
    srs_max = max( abs(x_neg) );
end
srs_min = min(x_pos);
if min( abs(x_neg) ) < srs_min
    srs_min = min( abs(x_neg) );
end
%
plot(fn,x_pos,fn,abs(x_neg),'-');
%
if iunit==1
    ylabel('Velocity (in/sec)');
else
    ylabel('Velocity (m/sec)');
end
xlabel('Natural Frequency (Hz)');
Q=1./(2.*damp);
out5 = sprintf(' Pseudo Velocity Shock Response Spectrum Q=%g ',Q);
title(out5);
grid;
set(gca,'MinorGridLineStyle','none','GridLineStyle',':','XScale','log','YScale','log');
legend('positive','negative',2);
%
ymax= 10^(round(log10(srs_max)+0.8));
ymin= 10^(round(log10(srs_min)-0.6));
%
axis([fmin,fmax,ymin,ymax]);
end
```