LOWESS- Locally Weighted Scatterplot Smoothing

Table of Contents

Modifications:

This regression will work on linear and non-linear relationships between X and Y.

```
12/19/2008 - added upper and lower LOWESS smooths.  These additional
smooths show how the distribution of Y varies with X.  These smooths
are simply LOWESS applied to the positive and negative residuals
separately, then added to the original lowess of the data.  The same
smoothing factor is applied to both the upper and lower limits.


2/21/2009 - added sorting to the function, data no longer need to be
sorted.  Also added a routine such that if a user also supplies a
second dataset, linear interpolations are done one the lowess and used
to predict y-values for the supplied x-values.


10/22/2009 - modified the second user provided X-data for obtaining
predictions.  Matlab function unique sorts by default.  It really was
not needed in the section of code to perform linear interpolations of
the x-data using the y-predicted LOWESS results.  If the user does not
supply a second x-data set, it will assume to use the supplied x-y
data set.  Thus there is an output (xy) that maintains the original
sequence of the input.  Additionally, the user can now include a
sequence index as the first column of input data. This can be a
datenum or some other ordering index.  The output will be sequenced
using that index.  If a sequence index is provided a second subplot
will be created show the predicted Y-values in the order of the
included sequence index. I suspect this sequence index most often will
be a DateTime (i.e. datenum).  Just to the function generic enough,
the X-axis labels are not converted to a nice date format, but the
user could easily change that with a datetic attribute in the subplot.


An example dataset (skl1a.mat) is also included in the ZIP file for
convenience.
```

Description

Using a robust regression like LOWESS allows one the ability to detect a trend in data that may otherwise have too much variance resulting in non-significance p-values.

Yhat (prediction) is computed from a weghted least squares regression whose weights are both a function of distance from X and magnitude from of the residual from the previous regression.

The conceptual of these functions and subfunctions follow the USGS Kendall.exe routines. Because matlab is 8-byte precision, there are some very small differences between FORTRAN compiled and matlab. Maybe 64-bit OS's has 16-byte precision in matlab?

There is a very simple subfunction to create a plot of the data and regression if the user so choses with a flag in the call to the lowess function. BTW-- the png file looks much better than what the figure looks like on screen.

There are loops in these routines to keep the memory requirements to a minimum, since it is foreseeable that one may have very large datasets to work with.

f = a smoothing factor between 0 and 1. The closer to one, the more smoothing done.

Syntax:

```
[dataout lowerLimit upperLimit xy] = lowess(datain,f,wantplot,imagefile,xdata)


datain = n x 2 (or 3 if sequend index is included) matrix
dataout = n x 3 matrix
wantplot = scaler (optional)
     if ~= 0 then create plot
imagefile = full path and file name where to output the figure to an
     png file type at 600 dpi. If imagefile not provided, a figure will
     be displayed but not exported to a graphics file.
     e.g. imagefile = 'd:\temp\lowess.png';
xdata = n x 1 vector. The user can supply a second dataset of x-values
     that will be used to predict y-values using the lowess regression
     results.
xy = x-values supplied by the user in xdata (or taken from the input
     data), and y-predictions using the lowess regression results.  If
     a sequence index is given this will be included as well and
     inserted as the first column.
```

where:

```
*  datain(:,1) = x
*  datain(:,2) = y
*  f = scaler (0 < f < 1)
*  wantplot = scaler
*  imagefile = string


*  dataout(:,1) = x
*  dataout(:,2) = y
*  dataout(:,3) = y-prediction (aka yhat)
*  lowerLimit(:,1) = x with negative residuals
*  lowerLimit(:,2) = y-prediction of residuals + original y-prediction
*  upperLimit(:,1) = x with positive residuals
*  upperLimit(:,2) = y-prediction of residuals + original y-prediction
```

Requirements: none

Written by

Jeff Burkey
King County Department of Natural Resources and Parks
email: jeff.burkey@kingcounty.gov
12/16/2008

Example syntax: [dataout lowerLimit upperLimit xy] = lowess(skl1a,.25,1,'c:\temp\test.png',xdata)

Primary Function: lowess

The main engine for this function.

```
function [dataout lowerLimit upperLimit xy] = lowess(datain,f,wantplot,imagefile,x


    % start timer
    start = tic;

    rowcol = size(datain);

    if rowcol(2) == 3
        % assume time index for first column
        dte = datain(:,1);
        x_data = datain(:,2);
        y_data = datain(:,3);
    else % assign empty set
        dte = [];
        x_data = datain(:,1);
        y_data = datain(:,2);
    end

    if exist('xdata','var') == 0
        % User didn't provide any x-values for generating a dataset use
        % supplied set prior to sorting.
        xdata = x_data;
    end

    datain = sortrows([x_data y_data]);

    if exist('wantplot','var') == 0 || wantplot == 0
        % user didn't provide assume zero (i.e. no plot)
        fprintf('\nNo plot will be created.\n');
        wantplot = 0;
        imagefile = '';
        limits = 1;
        upperLimit = nan;
        lowerLimit = nan;
    else
        limits = 3;
    end
    if exist('imagefile','var') == 0
        % User didn't provide do not export to graphics file
        fprintf('\nNo plot will be exported.\n');
        imagefile = '';
    end

    dataout = [];

    for nplots=1:limits
        % if limits is turned on, then plot the upper and lower limits of
        % the lowess- set to plot residuals lowess
        row = find(datain(:,1));
```

```
x = datain(row,1);
y = datain(row,2);

switch nplots
    case 2
        row = lwsResiduals > 0;
        x = dataout(row,1);
        y = lwsResiduals(row);
    case 3
        row = lwsResiduals < 0;
        x = dataout(row,1);
        y = lwsResiduals(row);
end

n = length(x);

if (f <= 0.0)
    f=0.25; % set to default
end

m=fix(n*f+0.5);
window = zeros(n,1);
yhat = zeros(n,1);

for j=1:n
    % This could be done in a matrix, but need to keep memory footprint
    % small, thus the loop.
    d = abs(x- x(j));
    r1 = ones(n,1);
    d = sort(d);

    window(j)=d(m);
    yhat(j)= rwlreg(x,y,n,window(j),r1,x(j));
end

for it=1:2
    e = abs(y-yhat);

    n = length(e);
    s=median(e);

    r = e/(6*s);
    r = 1-r.^2;
    r = max(0.d0,r);
    r = r.^2;

    for j=1:n
        yhat(j)= rwlreg(x,y,n,window(j),r,x(j));
    end
end

switch nplots
    case 1
        % calculate residuals otherwise skip
        lwsResiduals = y - yhat;
        dataout = [x y yhat];
    case 2
        ul = [x y yhat];
        [~, ia, ib] = intersect(dataout(:,1),ul(:,1));
        upperLimit = [ul(ib,1) ul(ib,3) + dataout(ia,3)];
        clear ul c ia ib
    case 3
        ll = [x y yhat];
        [~, ia, ib] = intersect(dataout(:,1),ll(:,1));
        lowerLimit = [ll(ib,1) ll(ib,3) + dataout(ia,3)];
```

```
                clear ul c ia ib
        end
    end

    fprintf('\nCompute time %6.4f seconds.\n',toc(start));
```

Generate predicted XY data

Using linear interpolation to estimate y from the lowess regression Any x-values beyond the range given to generate the lowess will be ignorged. Matlab unique function sorts the data, thus a modified unique function (usunique) was developed to return an unsorted vector.

```
    if xdata
        xyd = [dataout(:,1) dataout(:,3)];
        xyd = unique(xyd,'rows');
        xd = xdata(xdata >= min(xyd(:,1)));
        xd = xd(xd <= max(xyd(:,1)));
        if ~isnan(xyd)
            % if f value was too small, it's possible that multiple

            % Note:  it may be possible to have a few nan's in the data set
            % while the results would still be valid.  I haven't come
            % across a case of this but is possible.  If so, then the user
            % may need to incorporate a threshold of just how many nan's
            % would be acceptable before dumping the whole regression.  But
            % to be conservative, if there are any nan's throw out the
            % whole result dataset.
            yinterp = interp1(xyd(:,1),xyd(:,2),xd);

            xy = [dte xd yinterp];

            if length(xd) ~= length(xdata)
                fprintf('\nOne or more x-values were beyond the range supplied to
            end
        else
            fprintf('\nThere were NaNs in the lowess results. No plot will be crea
            wantplot = 0;
            xy = nan;
        end
    end

    if wantplot ~= 0
        customplot(dataout,upperLimit,lowerLimit,f,[dte x_data y_data],xy,imagefil
    end


end
```

Modification of check for 10 or more non-zero weights

by Hirsch June 1987

Robust weighted least squares regression, bisquare weights by distance on X-axis. x = is the estimation point yy = is the estimate value of y at x dd = is half the width of the window r = is the robustness weight, a bisquare weight of residuals.

```
function [yy] = rwlreg(x,y,n,d,r,xx)
```

```
    dd=d;
    ddmax = abs(x(n) - x(1));
    if dd == 0.0
        error('Regression:lowess','LOWESS window size = 0. Increase f.');
    else
        while dd <= ddmax
            c = 0;
            total = 0.0;
            f = (abs(x-xx)/dd);
            f = 1.0-f.^3;
            w = ((max(0.d0,f)).^3).*r;
            total = sum(w);
            c = sum(w>0);
            if c > 3
                break % out of while loop
            else
                dd=1.28*dd;
                fprintf('\nrobust size of window = %5.0f.\nLowess window size incr
            end
        end
    end

    w = w/total;

    [a b] = wlsq(x,y,w);
    yy=a+b*xx;
end
```

Weighted least squares

This subfunction does not require any toolboxes in matlab to execute.

```
function[a b] = wlsq(x,y,w)
    sumw = abs(1-sum(w));
    if sumw > 1e-10
        % The weights, w, must sum to one. Precision assuming type double,
        %    The user may want to adjust this value.
        error('Regression:wlsq','\nThere is an error in the weights.\nWeights do n
    end
    wxx = sum(w.*x.^2);
    wx = sum(w.*x);
    wxy = sum(w.*x.*y);
    wy = sum(w.*y);
    b = (wxy-wy*wx)/(wxx-wx^2);
    a = wy-b*wx;
end
```

Compute time 5.7452 seconds.

Plotting of data and lowess regression line

If a sequence vector is included in the data, the figure will contain two subplots. The first one is the LOWESS regression of the data, the second plots the time in the original sequence using the first column of input data. Example a datenum for when the data were observed would be common. The second plot will plot the observed Y-data and the predicted Y-data.

```
function customplot(lws,uplmt,lwrlmt,f,oldxy,newxy,imgfile)
    figure1 = figure;
```

```
    try
        rowcol = size(newxy);
        if rowcol(2) == 3 % Users provided a sequence index e.g. Datenum
            % The second subplot id defined first as a matter of
            % readability in the code.  This Conditional segment will not
            % be executed if no sequence index is provided (e.g. datetime).
            subplot(2,2,3:4,'Parent',figure1,...
                'YScale','linear','YMinorTick','off',...
                'XScale','linear','XMinorTick','on',...
                'YMinorGrid','off',...
                'XMinorTick','on',...
                'XMinorGrid','on');
            box('on');
            grid('on');
            hold('all');

            line(oldxy(:,1),oldxy(:,3),'LineStyle','none', ...
                'Marker','o','MarkerSize',7,...
                'MarkerEdgeColor','k',...
                'MarkerFaceColor','b',...
                'DisplayName','Observed');

            line(newxy(:,1),newxy(:,3),'LineStyle','-', ...
                'LineWidth',2,'Color','r',...
                'DisplayName','Simulated');

            ylabel('y-Values');
            xlabel('Sequence Index (e.g. datenum)');
            title('Simulated y-Values using LOWESS Regression');
            hold('off');

            % This is the primary plot that will be generate either from
            % this conditional statement or in the ELSEIF below. They are
            % exact same except for defining Subplot space as either two
            % plots or one.
            subplot(2,2,[1 2],'Parent',figure1,...
                'YScale','linear','YMinorTick','off',...
                'XScale','linear','XMinorTick','on',...
                'YMinorGrid','off',...
                'XMinorTick','on',...
                'XMinorGrid','on');

        elseif rowcol(2) == 2
            % No sequence index is given, second plot would be identical to
            % first plot.  Define plot to occupy space of both subplots.
            % This could be revised and not even call it a subplot, but for
            % consistency it is.
            subplot(2,1,[1 2],'Parent',figure1,...
                'YScale','linear','YMinorTick','off',...
                'XScale','linear','XMinorTick','on',...
                'YMinorGrid','off',...
                'XMinorTick','on',...
                'XMinorGrid','on');
        end

        box('on');
        grid('on');
        hold('all');

        x = lws(:,1);
        y = lws(:,2);
        yh = lws(:,3);

        % Point plot of points of the observed data on the LOWESS plot
```

```
        line(x,y,'LineStyle','none', ...
            'Marker','o','MarkerSize',7,...
            'MarkerEdgeColor','k',...
            'MarkerFaceColor','b',...
            'DisplayName','Data');

        % Line plot of the LOWESS regression
        line(x,yh, 'Color','r', 'LineWidth',2, 'LineStyle','-')


        x = uplmt(:,1);
        yh = uplmt(:,2);

        % Line plot of the upper limit LOWESS regression
        line(x,yh, 'Color', 'r', 'LineWidth',2,'LineStyle',':')

        x = lwrlmt(:,1);
        yh = lwrlmt(:,2);

        % Line plot of the lower limit LOWESS regression
        line(x,yh, 'Color', 'r', 'LineWidth',2,'LineStyle',':')

        grid on
        xlabel('x-Values')
        ylabel('y-Values')
        ts = strcat('LOWESS Regression plot f=',num2str(f));
        title(ts)

        hold('off');

        if ~isempty(imgfile)
            % If a filename is give for the plot, create a PNG file.
            fprintf('\nCreating plot. Give a few tics.\n');
            print('-dpng','-r100', imgfile);
            fprintf('\nFinished...\n');
        end

        close(figure1)
    catch ME1
        disp(ME1)
    end
end
```

```
Creating plot. Give a few tics.

Finished...

dataout =

  1.0e+003 *

   0.220000000000000   3.910000000000000   4.319232343949838
   0.220000000000000   4.630000000000000   4.319232343949838
   0.230000000000000   1.640000000000000   4.071345778164299
   0.230000000000000   2.340000000000000   4.071345778164299
   0.250000000000000   3.810000000000000   3.576523856445645
   0.250000000000000   4.930000000000000   3.576523856445645
   0.260000000000000   4.590000000000000   3.330649521694364
   0.260000000000000   4.670000000000000   3.330649521694364
   0.280000000000000   1.260000000000000   2.843146723319974
   0.310000000000000   7.640000000000000   2.081088886040460
   0.320000000000000   1.500000000000000   1.776854439448522
   0.320000000000000   3.070000000000000   1.776854439448522
```

0.330000000000000	1.400000000000000	1.467806032861281
0.330000000000000	1.500000000000000	1.467806032861281
0.350000000000000	1.020000000000000	1.085764453022432
0.350000000000000	1.080000000000000	1.085764453022432
0.360000000000000	1.060000000000000	0.949472454437612
0.370000000000000	0.595000000000000	0.828002843257433
0.370000000000000	0.741000000000000	0.828002843257433
0.380000000000000	0.777000000000000	0.707819038945072
0.390000000000000	0.321000000000000	0.644857298457643
0.390000000000000	0.423000000000000	0.644857298457643
0.390000000000000	0.472000000000000	0.644857298457643
0.390000000000000	0.542000000000000	0.644857298457643
0.400000000000000	1.080000000000000	0.579378508781684
0.410000000000000	0.366000000000000	0.576968965609245
0.410000000000000	0.508000000000000	0.576968965609245
0.410000000000000	0.542000000000000	0.576968965609245
0.410000000000000	0.569000000000000	0.576968965609245
0.420000000000000	0.495000000000000	0.575254719695504
0.430000000000000	0.565000000000000	0.581089968351493
0.430000000000000	0.665000000000000	0.581089968351493
0.430000000000000	0.679000000000000	0.581089968351493
0.430000000000000	0.680000000000000	0.581089968351493
0.430000000000000	0.736000000000000	0.581089968351493
0.430000000000000	0.920000000000000	0.581089968351493
0.440000000000000	0.460000000000000	0.551199460990149
0.440000000000000	1.230000000000000	0.551199460990149
0.450000000000000	0.380000000000000	0.505773692628709
0.450000000000000	0.469000000000000	0.505773692628709
0.450000000000000	0.469000000000000	0.505773692628709
0.450000000000000	0.541000000000000	0.505773692628709
0.460000000000000	0.264000000000000	0.450226907297555
0.460000000000000	0.295000000000000	0.450226907297555
0.460000000000000	0.364000000000000	0.450226907297555
0.460000000000000	0.490000000000000	0.450226907297555
0.470000000000000	0.239000000000000	0.421978673193178
0.470000000000000	0.462000000000000	0.421978673193178
0.470000000000000	0.499000000000000	0.421978673193178
0.470000000000000	0.750000000000000	0.421978673193178
0.480000000000000	0.334000000000000	0.411046051508763
0.480000000000000	0.342000000000000	0.411046051508763
0.490000000000000	0.458000000000000	0.418143316553890
0.490000000000000	0.488000000000000	0.418143316553890
0.490000000000000	0.500000000000000	0.418143316553890
0.500000000000000	0.216000000000000	0.433178692395648
0.500000000000000	0.244000000000000	0.433178692395648
0.500000000000000	0.314000000000000	0.433178692395648
0.500000000000000	0.473000000000000	0.433178692395648
0.500000000000000	0.503000000000000	0.433178692395648
0.510000000000000	0.223000000000000	0.441858831468175
0.510000000000000	0.593000000000000	0.441858831468175
0.520000000000000	0.472000000000000	0.435657636954711
0.520000000000000	0.732000000000000	0.435657636954711
0.530000000000000	0.583000000000000	0.422193300564489
0.540000000000000	0.360000000000000	0.408850209538940
0.540000000000000	0.469000000000000	0.408850209538940
0.550000000000000	0.245000000000000	0.399199260547102
0.550000000000000	0.513000000000000	0.399199260547102
0.560000000000000	0.444000000000000	0.403253382069620
0.570000000000000	0.224000000000000	0.405156030562639
0.580000000000000	0.325000000000000	0.402782997047014
0.620000000000000	0.240000000000000	0.396157918194194
0.620000000000000	0.432000000000000	0.396157918194194
0.620000000000000	0.650000000000000	0.396157918194194
0.670000000000000	0.279000000000000	0.384372251771626

```
   0.680000000000000   0.533000000000000   0.382404863909111
   0.700000000000000   0.266000000000000   0.378608820106509
   0.700000000000000   0.578000000000000   0.378608820106509
   0.710000000000000   0.245000000000000   0.376631059080293


lowerLimit =

  1.0e+003 *

   0.220000000000000   2.522921607626112
   0.230000000000000   2.414996556268145
   0.280000000000000   1.873587116875478
   0.320000000000000   1.717432442097954
   0.330000000000000   1.399186245213494
   0.350000000000000   0.989297425343045
   0.370000000000000   0.705759366232837
   0.390000000000000   0.505090628398852
   0.410000000000000   0.479796992537626
   0.420000000000000   0.497941052781938
   0.430000000000000   0.507659003913640
   0.440000000000000   0.469824864468880
   0.450000000000000   0.412982217677612
   0.460000000000000   0.343066681325003
   0.470000000000000   0.303358779245056
   0.480000000000000   0.272288533965245
   0.500000000000000   0.268632046646991
   0.510000000000000   0.292980429348981
   0.540000000000000   0.268183623676995
   0.550000000000000   0.262275549522692
   0.570000000000000   0.269305837186851
   0.580000000000000   0.268535699815678
   0.620000000000000   0.268453691745118
   0.670000000000000   0.261669609211342
   0.700000000000000   0.259042712320433
   0.710000000000000   0.258118060931821


upperLimit =

  1.0e+003 *

   0.220000000000000   5.924703215376282
   0.250000000000000   4.958214149638528
   0.260000000000000   4.634423450397931
   0.310000000000000   2.968233923168393
   0.320000000000000   2.578400061934314
   0.330000000000000   2.191836508167800
   0.360000000000000   1.465036229387142
   0.380000000000000   0.901231487628006
   0.400000000000000   0.756327037978211
   0.430000000000000   0.740842405028191
   0.440000000000000   0.676007139348845
   0.450000000000000   0.612084254582316
   0.460000000000000   0.516519141991478
   0.470000000000000   0.500509400861237
   0.490000000000000   0.502378113582879
   0.500000000000000   0.525519837997631
   0.510000000000000   0.550215675502173
   0.520000000000000   0.543920245613710
   0.530000000000000   0.525649091771840
   0.540000000000000   0.508470364367557
   0.550000000000000   0.488134853199903
   0.560000000000000   0.483487758959327
```

```
   0.620000000000000   0.529206591147893
   0.680000000000000   0.554311265850982
   0.700000000000000   0.563297949607988


xy =

  1.0e+005 *

   7.210040000000000   0.004900000000000   0.004181433165539
   7.210350000000000   0.005400000000000   0.004088502095389
   7.210940000000000   0.002200000000000   0.043192323439498
   7.211850000000000   0.003900000000000   0.006448572984576
   7.212770000000000   0.004500000000000   0.005057736926287
   7.213690000000000   0.002300000000000   0.040713457781643
   7.214590000000000   0.003600000000000   0.009494724544376
   7.215500000000000   0.004600000000000   0.004502269072976
   7.216420000000000   0.004300000000000   0.005810899683515
   7.217340000000000   0.004300000000000   0.005810899683515
   7.218250000000000   0.006200000000000   0.003961579181942
   7.219470000000000   0.004600000000000   0.004502269072976
   7.220080000000000   0.004500000000000   0.005057736926287
   7.221000000000000   0.005800000000000   0.004027829970470
   7.221900000000000   0.003500000000000   0.010857644530224
   7.222810000000000   0.004400000000000   0.005511994609901
   7.223730000000000   0.005300000000000   0.004221933005645
   7.224040000000000   0.003800000000000   0.007078190389451
   7.224340000000000   0.004400000000000   0.005511994609901
   7.224650000000000   0.004300000000000   0.005810899683515
   7.224960000000000   0.006800000000000   0.003824048639091
   7.225240000000000   0.002500000000000   0.035765238564456
   7.225550000000000   0.002500000000000   0.035765238564456
   7.225850000000000   0.004500000000000   0.005057736926287
   7.226160000000000   0.005000000000000   0.004331786923956
   7.226460000000000   0.005100000000000   0.004418588314682
   7.226770000000000   0.004900000000000   0.004181433165539
   7.227080000000000   0.007000000000000   0.003786088201065
   7.227380000000000   0.004200000000000   0.005752547196955
   7.227690000000000   0.007100000000000   0.003766310590803
   7.227990000000000   0.004300000000000   0.005810899683515
   7.228300000000000   0.004100000000000   0.005769689656092
   7.228610000000000   0.007000000000000   0.003786088201065
   7.228890000000000   0.002600000000000   0.033306495216944
   7.229200000000000   0.002600000000000   0.033306495216944
   7.229500000000000   0.005000000000000   0.004331786923956
   7.229810000000000   0.004500000000000   0.005057736926287
   7.230110000000000   0.005000000000000   0.004331786923956
   7.230420000000000   0.006200000000000   0.003961579181942
   7.230730000000000   0.006700000000000   0.003843722517716
   7.231030000000000   0.004100000000000   0.005769689656092
   7.231340000000000   0.004700000000000   0.004219786731932
   7.231640000000000   0.003700000000000   0.008280028432574
   7.231950000000000   0.004100000000000   0.005769689656092
   7.232260000000000   0.005400000000000   0.004088502095389
   7.232550000000000   0.005500000000000   0.003991992605471
   7.232860000000000   0.002200000000000   0.043192323439498
   7.233160000000000   0.004600000000000   0.004502269072976
   7.233470000000000   0.003900000000000   0.006448572984576
   7.233770000000000   0.005500000000000   0.003991992605471
   7.234080000000000   0.003200000000000   0.017768544394485
   7.234390000000000   0.005700000000000   0.004051560305626
   7.234690000000000   0.004800000000000   0.004110460515088
   7.235300000000000   0.005200000000000   0.004356576369547
   7.235610000000000   0.006200000000000   0.003961579181942
```

7.235920000000000	0.005200000000000	0.004356576369547
7.236200000000000	0.004300000000000	0.005810899683515
7.236510000000000	0.004000000000000	0.005793785087817
7.236810000000000	0.004300000000000	0.005810899683515
7.237120000000000	0.004900000000000	0.004181433165539
7.237420000000000	0.005600000000000	0.004032533820696
7.237730000000000	0.003700000000000	0.008280028432574
7.238040000000000	0.004600000000000	0.004502269072976
7.238340000000000	0.003900000000000	0.006448572984576
7.238950000000000	0.003300000000000	0.014678060328613
7.239850000000000	0.003500000000000	0.010857644530224
7.240460000000000	0.004800000000000	0.004110460515088
7.240770000000000	0.003900000000000	0.006448572984576
7.241380000000000	0.005000000000000	0.004331786923956
7.242300000000000	0.004100000000000	0.005769689656092
7.243220000000000	0.004700000000000	0.004219786731932
7.244110000000000	0.002800000000000	0.028431467233200
7.245030000000000	0.005100000000000	0.004418588314682
7.245950000000000	0.004700000000000	0.004219786731932
7.246870000000000	0.003100000000000	0.020810888860405
7.247770000000000	0.002300000000000	0.040713457781643
7.248380000000000	0.004700000000000	0.004219786731932
7.249610000000000	0.003300000000000	0.014678060328613
7.250810000000000	0.003200000000000	0.017768544394485
7.251420000000000	0.005000000000000	0.004331786923956

Published with MATLAB® 7.9