Table of Contents

The results are exactly same with Bloomberg and Dr. K's Final Excel

```
% 2 refers the user wants simple moving average
% 1.9 = sigma
% 30= length
% 0.94 = lambda
[profits1,cumulativeprofit1,short_prices1,long_prices1]=...
 bollingerband1d(2,1.9,30,0.94)
% If we play with numbers
[profits2,cumulativeprofit2,short_prices2,long_prices2]= ...
 bollingerband1d(2,2.5,60,0.94)

% 1 refers the user wants exponentially weighted moving average
% 1.9 = sigma
% 30= length
% 0.94 = lambda
   [profits3,cumulativeprofit3,short_prices3,long_prices3]=...
       bollingerband1d(1,1.9,30,0.94)
% If we play with numbers
   [profits4,cumulativeprofit4,short_prices4,long_prices4]=...
       bollingerband1d(1,2.5,60,0.94)


% We can play with numbers till morning. But what if we create a for
 loop
% which will show us the best numbers?????

% first number 2 refers simple moving average
%  1.9= where the sigma starts
%  30= where the length starts
%  0.94=lambda
[cumulativeprofit5]=bollingerband3d(2,1.9,30,0.94)


% First number 1 refers exponentially weighted moving average
%  1.9= where the sigma starts
%  30= where the length starts
%  0.94=lambda
[cumulativeprofit6]=bollingerband3d(1,1.9,30,0.94)
```

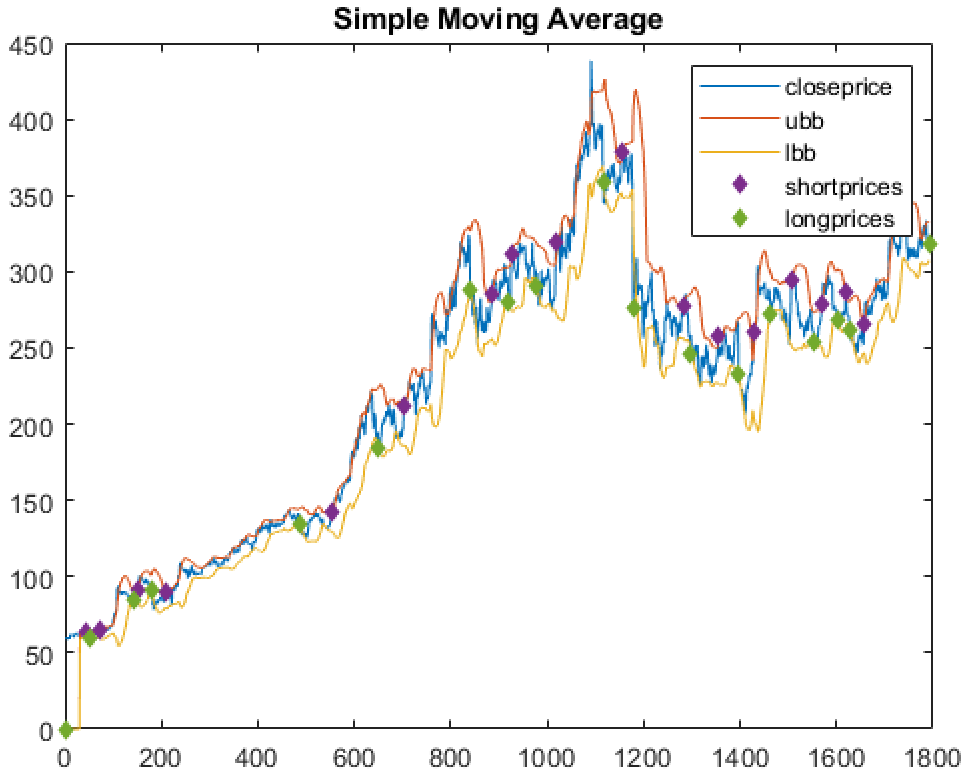
1d shows where to short or long and profits: also scatter them in a graph

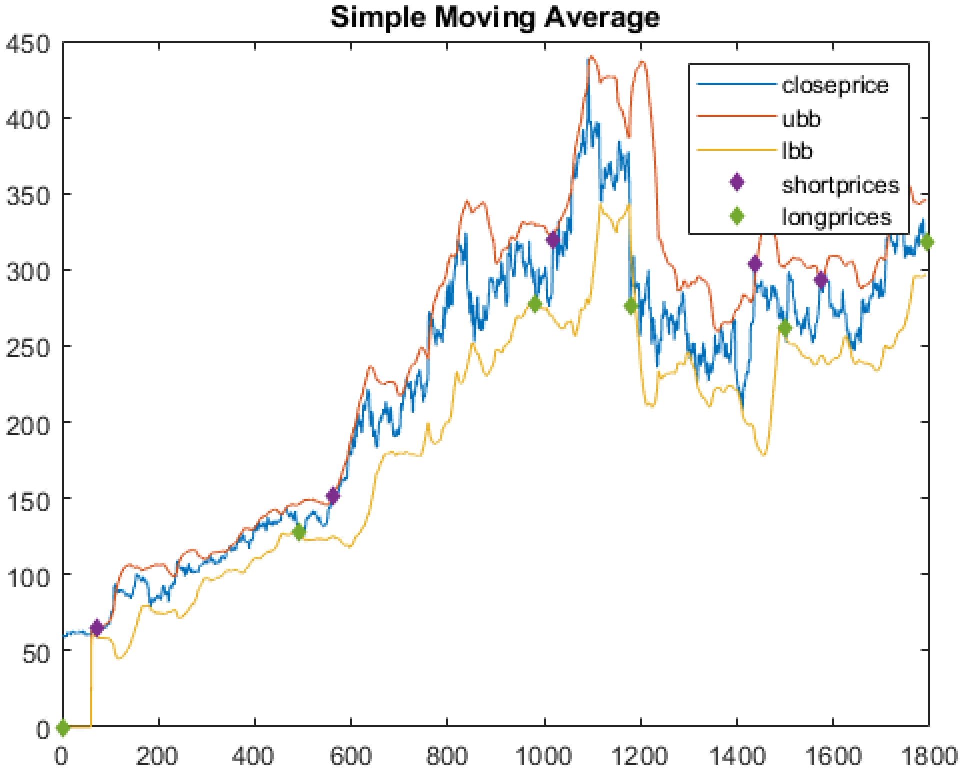
```
function [profits,cumulativeprofit,short_prices,long_prices]=...
    bollingerband1d(ExpWeighOrSimp,sigma,lengthh,lambda)
   %ExpWeighOrSimp=1 exponential weighted moving average
   %ExpWeighOrSimp=2 simple moving average
load('price.mat');
endpoint=size(price,1);

if ExpWeighOrSimp==1 %finding Ewma if the user wants exponential mov.
 avrg.
    weight=zeros(endpoint,1);
    for i=1:lengthh
        weight(lengthh+1-i,1)=(1-lambda)*lambda^(i-1);
    end
    stds=price.*weight;
    ewma(lengthh+1,1)=sum(stds(1:30));


    for i=lengthh+2:endpoint
        ewma(i,1)=(lambda*ewma(i-1,1))+((1-lambda)*price(i-1,1));
    end
end


     for i=1:endpoint-lengthh %Finding upper bounds and lower bounds
         if ExpWeighOrSimp==1
              lbb(lengthh+i,1)=ewma(lengthh+i,1)-(sigma*(std(price...
                  (i+1:i+lengthh)*((lengthh-1)^0.5))/
((lengthh-0)^0.5)));
         ubb(lengthh+i,1)=ewma(lengthh+i,1)+(sigma*(std(price...
             (i+1:i+lengthh)*((lengthh-1)^0.5))/((lengthh-0)^0.5)));
         else
         lbb(lengthh+i,1)=mean(price(i+1:i+lengthh))-(sigma*(std...
             (price(i+1:i+lengthh)*((lengthh-1)^0.5))/
((lengthh-0)^0.5)));
         ubb(lengthh+i,1)=mean(price(i+1:i+lengthh))+(sigma*(std...
             (price(i+1:i+lengthh)*((lengthh-1)^0.5))/
((lengthh-0)^0.5)));
         end
         if price(lengthh+i,1)<lbb(lengthh+i,1) %find where short or
 long
             long(lengthh+i,1)=1;
         else
             long(lengthh+i,1)=0;
         end
         if price(lengthh+i,1)> ubb(lengthh+i,1)
             short(lengthh+i,1)=1;
         else
             short(lengthh+i,1)=0;
         end
     end

     totalmoney=100000000;

     next_long=find(long~=0, 1, 'first');
     next_short=find(short~=0, 1, 'first');

     profit(next_long,1)= (price(next_short,1)-price(next_long,1))...
         *round(totalmoney/price(next_short,1),0);
     totalmoney=100000000+sum(profit);
     numberoflong=1;
     numberofshort=1;
     for  a=1:1000%Just a big number to make sure that covers all the
 array
       if i~=endpoint
         for i= next_long:endpoint
             if short(i,1)==1
                 next_short=i;
                 numberofshort=numberofshort+1;
                 break
             end
         end
         if i~=endpoint
             profit(next_short,1)= -1* ((price(next_long,1)-price...
                 (next_short,1))*round(totalmoney/
price(next_long,1),0));
             totalmoney=100000000+sum(profit);
         end
         for i= next_short:endpoint
             if long(i,1)==1
                 next_long=i;
                 numberoflong=numberoflong+1;
                 break
             end
         end
         if i~=endpoint
             profit(next_long,1)= (price(next_short,1)-price...
                 (next_long,1))*round(totalmoney/
price(next_short,1),0);
             totalmoney=100000000+sum(profit);
         end
       end
     end

     %Find MTM to cover the last trade
     if mod(numberoflong+numberofshort,2)==1
         if next_short>next_long
             profit(endpoint,1)= (price(next_short,1)-price...
                 (endpoint,1))*round(totalmoney/
price(next_short,1),0);
         end
         if next_long>next_short
             profit(endpoint,1)= -1* ((price(next_long,1)-price...
                 (endpoint,1))*round(totalmoney/
price(next_long,1),0));
         end
     end

     totalmoney=100000000+sum(profit);
     cumulativeprofit= sum(profit);
     [x,y,profits]=find(profit);

     first_long=find(long~=0, 1, 'first');
     first_short=find(short~=0, 1, 'first');

     shortprices=zeros(endpoint,1);
     longprices=zeros(endpoint,1);
     shortprices(first_short,1)= price(first_short,1);
      for j=1:2:(numberoflong+numberofshort)
      longprices(x(j),1)=price(x(j));
      end
      for j=2:2:(numberoflong+numberofshort)
          shortprices(x(j),1)=price(x(j));
      end
% this step is only for using scatter in the graph( there must be an
 easy
% way but this way tought me many other things.
[s1,s2,short_prices]=find(shortprices);
position1=zeros(endpoint,1);
sizet1=size(s1,1);
      for i=1:sizet1
      position1(s1(i),1)=s1(i);
      end
 [l1,l2,long_prices]=find(longprices);
position2=zeros(endpoint,1);
sizet2=size(l1,1);
      for i=1:sizet2
      position2(l1(i),1)=l1(i);
      end
figure
plot(price)
hold on
plot(ubb)
hold on
plot(lbb)
hold on
scatter(position1,shortprices,'filled','d')
hold on
scatter(position2,longprices,'filled','d')
legend('closeprice','ubb','lbb','shortprices','longprices')
if  ExpWeighOrSimp==2
title('Simple Moving Average')
else
    title('Exponentially Weighted Moving Average')
end
%     My results are checked in Bloomberg and all the results are
 correct..
```

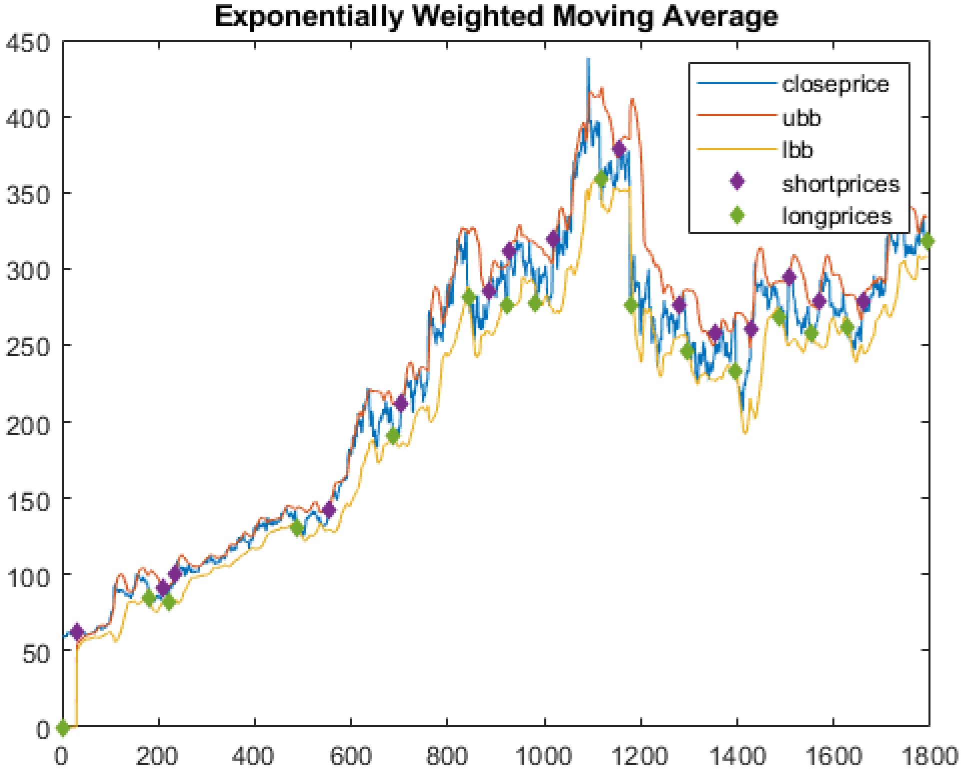
```
end

profits1 =

   1.0e+07 *

    0.5379
    0.8259
   -3.4053
    0.7207
    0.0602
   -0.0831
   -4.2299
    0.2514
   -1.3877
    0.5111
   -1.3521
   -0.0222
    0.0473
    0.2758
    0.1766
    0.2911
   -0.3885
    0.1583
    0.8070
    0.0094
    0.4240
    0.1949
    0.4226
    0.5670
   -0.2616
    0.4133
    0.7626
    0.6139
    0.2583
    0.4922
    0.6633
    0.1227
   -1.6754


cumulativeprofit1 =

  -3.1981e+07


short_prices1 =

   63.3530
   64.6430
   91.6220
   90.1210
  141.7760
  212.3920
  285.3300
  310.9640
  319.9350
  378.6190
  277.0400
  257.5120
  260.1650
  294.6230
  278.9100
  286.8900
  266.0000


long_prices1 =

   59.9450
   84.0140
   90.9860
  134.1580
  183.8330
  287.9360
  279.7750
  290.9940
  358.5570
  276.3580
  246.1640
  232.9100
  272.7380
  254.2420
  268.5300
  262.1500
  318.5700


profits2 =

   1.0e+07 *

   -9.6939
    0.0581
   -0.3041
    0.0091
    0.0094
    0.0079
    0.0122
    0.0119
   -0.0098


cumulativeprofit2 =
  -9.8993e+07


short_prices2 =

   64.6430
  151.4660
  319.9350
  304.0650
  292.6400


long_prices2 =

  127.3070
  277.9330
  276.3580
  261.2610
  318.5700


profits3 =

   1.0e+07 *

   -3.8031
    0.4586
    0.6990
    1.6436
   -2.7464
    0.5219
   -2.3311
    0.5088
   -1.6179
    0.0419
    0.1142
    0.4467
    0.4181
    0.6580
   -0.6051
    0.2466
    1.2570
   -0.0146
    0.6312
    0.3009
    0.6524
    0.8754
   -0.2790
    0.7743
    1.0933
    0.7964
    0.6454
    0.7145
   -1.7361
```

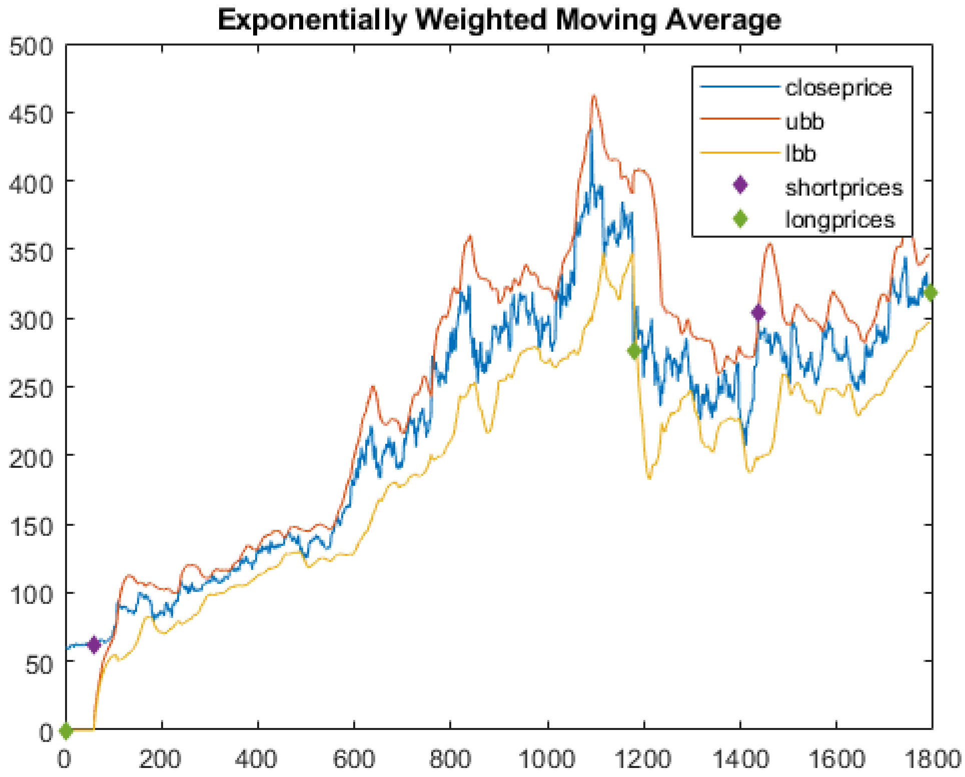
```
cumulativeprofit3 =

   3.6491e+06


short_prices3 =

   61.7600
   91.5570
  100.2530
  141.7760
  212.3920
  285.3300
  310.9640
  319.9350
  378.6190
  275.6760
  257.5120
  260.1650
  294.6230
  278.9100
  278.6000


long_prices3 =

   85.2480
   81.9410
  130.8520
  190.5670
  281.7920
  275.6760
  277.9330
  358.5570
  276.3580
  246.1640
  232.9100
  268.8510
  258.2310
  262.1500
  318.5700


profits4 =

   1.0e+08 *

   -3.4601
   -0.2466
    0.1291
```

```
cumulativeprofit4 =

  -3.5776e+08


short_prices4 =

   61.9620
  304.0650


long_prices4 =

  276.3580
  318.5700
```




Simple Moving Average
450
400
350
300
250
200
150
100
50
0
0
200
400
600
800
1000
1200
1400
1600
1800
closeprice
ubb
lbb
shortprices
longprices


Exponentially Weighted Moving Average
450
400
350
300
250
200
150
100
50
0
0
200
400
600
800
1000
1200
1400
1600
1800
closeprice
ubb
lbb
shortprices
longprices



3d where we can choose best sigma and length combinations

```
function [cumulativeprofit]=bollingerband3d...
    (ExpWeighOrSimp,sigma,lengthh,lambda) %

load('price.mat');
endpoint=size(price,1);
lbb=zeros(endpoint,1);
short=zeros(endpoint,1);
long=zeros(endpoint,1);
profit=zeros(10000,100);
profitss=zeros(100,100);

if ExpWeighOrSimp==1 %it is whether Exponentially Weighted or Simple
    weight=zeros(endpoint,1);
    for i=1:lengthh
        weight(lengthh+1-i,1)=(1-lambda)*lambda^(i-1);
    end
    stds=price.*weight;
    ewma(lengthh+1,1)=sum(stds(1:30));


    for i=lengthh+2:endpoint
```

```
        ewma(i,1)=(lambda*ewma(i-1,1))+((1-lambda)*price(i-1,1));
    end
end

 for lengthh=30:90
   for sigma=1.9:0.1:3
     for i=1:endpoint-lengthh
         if ExpWeighOrSimp==2
             lbb(lengthh+i,1)=mean(price(i+1:i+lengthh))-
(sigma*(std...
                 (price(i+1:i+lengthh)*...
                 ((lengthh-1)^0.5))/((lengthh-0)^0.5)));
             ubb(lengthh+i,1)=mean(price(i+1:i+lengthh))+(sigma*...
                 (std(price(i+1:i+lengthh)*...
                 ((lengthh-1)^0.5))/((lengthh-0)^0.5)));
         else
         lbb(lengthh+i,1)=ewma(lengthh+i,1)-(sigma*(std(price...
             (i+1:i+lengthh)*((lengthh-1)^0.5))/((lengthh-0)^0.5)));
         ubb(lengthh+i,1)=ewma(lengthh+i,1)+(sigma*(std(price...
             (i+1:i+lengthh)*((lengthh-1)^0.5))/((lengthh-0)^0.5)));
         end

         if price(lengthh+i,1)<lbb(lengthh+i,1)
             long(lengthh+i,1)=1;
         else
             long(lengthh+i,1)=0;
         end
         if price(lengthh+i,1)> ubb(lengthh+i,1)
             short(lengthh+i,1)=1;
         else
             short(lengthh+i,1)=0;
         end
     end

     totalmoney(lengthh-29)=100000000;

     next_long=find(long~=0, 1, 'first');
     next_short=find(short~=0, 1, 'first');

     profit(next_long,lengthh-29)= (price(next_short,1)-price...
         (next_long,1))*round(totalmoney(lengthh-29)/
price(next_short,1),0);
     totalmoney(lengthh-29)=100000000+sum(profit(:,lengthh-29));
     numberoflong=1;
     numberofshort=1;
     for  a=1:1000%Just a big number to make sure that covers all the
 array
       if i~=endpoint
         for i= next_long:endpoint
             if short(i,1)==1
                 next_short=i;
                 numberofshort=numberofshort+1;
                 break
             end
         end
         if i~=endpoint
             profit(next_short,lengthh-29)= -1*
 ((price(next_long,1)...
                 -price(next_short,1))*round(totalmoney(lengthh-29)...
                 /price(next_long,1),0));

 totalmoney(lengthh-29)=100000000+sum(profit(:,lengthh-29));
         end
         for i= next_short:endpoint
             if long(i,1)==1
                 next_long=i;
                 numberoflong=numberoflong+1;
                 break
             end
         end
         if i~=endpoint
             profit(next_long,lengthh-29)= (price(next_short,1)-
price...
                 (next_long,1))*round(totalmoney(lengthh-29)/...
                 price(next_short,1),0);

 totalmoney(lengthh-29)=100000000+sum(profit(:,lengthh-29));
         end
       end
     end

     %Find MTM to cover the last trade
     if mod(numberoflong+numberofshort,2)==1
         if next_short>next_long
             profit(endpoint,lengthh-29)= (price(next_short,1)-
price...
                 (endpoint,1))*round(totalmoney(lengthh-29)/...
                 price(next_short,1),0);
         end
         if next_long>next_short
             profit(endpoint,lengthh-29)= -1* ((price(next_long,1)-
price...
                 (endpoint,1))*round(totalmoney(lengthh-29)/...
                 price(next_long,1),0));
         end
     end

     totalmoney(lengthh-29,(sigma*10)-18)=100000000+sum(profit...
         (:,lengthh-29));
     cumulativeprofit(lengthh-29,(sigma*10)-18)=
 sum(profit(:,lengthh-29));
     [x,y,profits]=find(profit(:,lengthh-29));

   end
 end

figure
surf(cumulativeprofit)
```

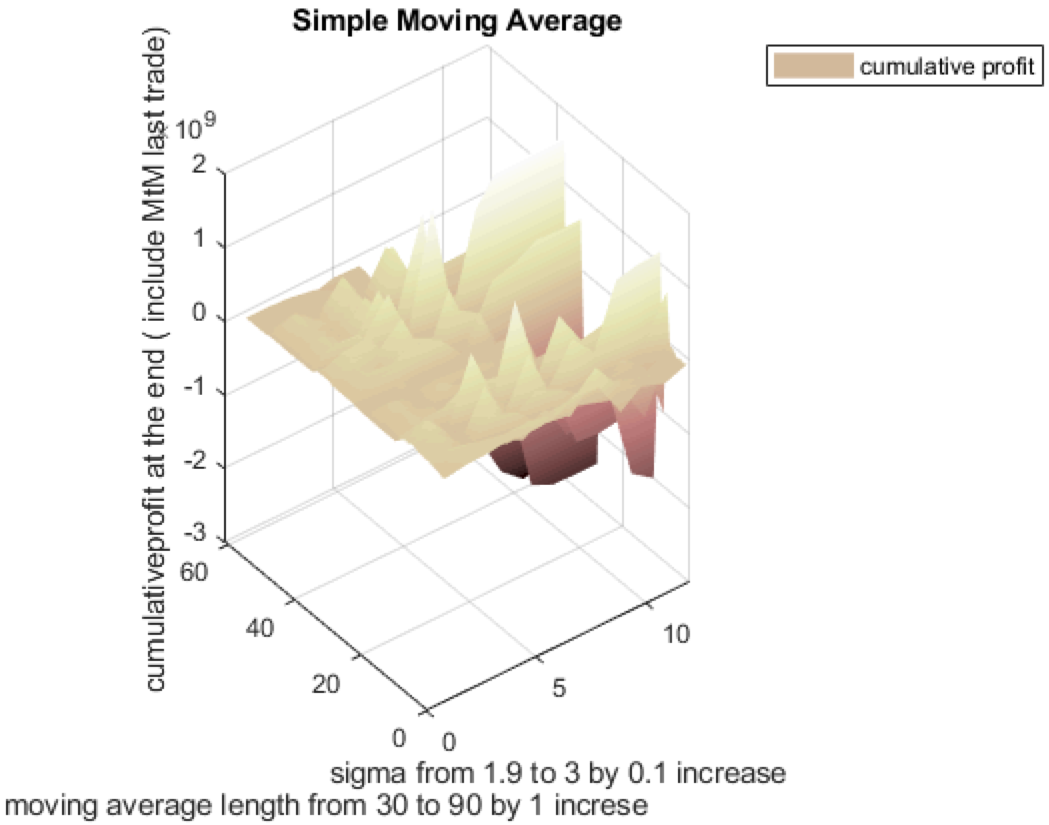
```
shading interp
colormap(pink)
legend('cumulative profit')
xlabel('sigma from 1.9 to 3 by 0.1 increase')
ylabel('moving average length from 30 to 90 by 1 increse')
zlabel('cumulativeprofit at the end ( include MtM last trade)')
if ExpWeighOrSimp==2
    title('Simple Moving Average')
else
    title('Exponentially Weighted Moving Average')
end

end
```

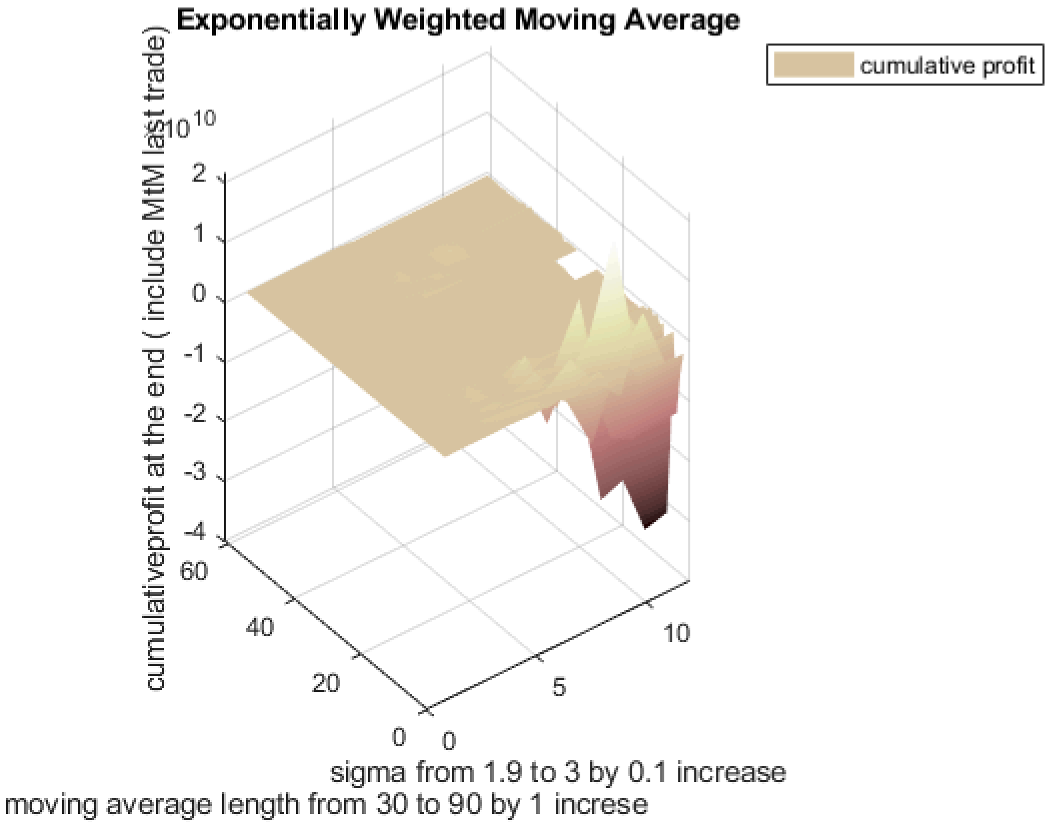
```
cumulativeprofit5 =

   1.0e+09 *

  Columns 1 through 7

   -0.0320    0.0169    0.0246    0.0741    0.0328    0.0553    0.2524
   -0.0433    0.0111    0.0833   -0.0297    0.1368    0.0609    0.1459
   -0.0199    0.0210    0.0172    0.0556    0.0437    0.0577   -0.1487
   -0.0106    0.0238    0.0679   -0.0011   -0.2066    0.0562   -0.1762
    0.0024    0.1010   -0.1952   -0.1005   -0.1361   -0.1819   -0.1106
    0.0188    0.0642   -0.0409   -0.1274    0.0939   -0.1752    0.1132
    0.0119    0.0297   -0.1210    0.1783   -0.1766    0.2091   -0.3224
    0.0270    0.0938   -0.1188    0.2753   -0.1043   -0.1614   -0.1062
    0.0281    0.1230   -0.1835    0.4880   -0.1481    0.6242   -0.4250
    0.0461    0.0965   -0.1984   -0.1350   -0.1112   -0.2256   -0.0511
    0.0642    0.1870   -0.1531   -0.1326   -0.0132   -0.2594   -0.0901
    0.0278    0.1993   -0.1467   -0.1531    0.0138   -0.1479    0.0064
    0.0426    0.1652   -0.1829    0.7861   -0.2173    1.2056   -0.6155
    0.0126    0.1316   -0.1838    0.3521   -0.1562    0.1967   -0.1215
    0.0017   -0.1177    0.1049   -0.0542    0.1959   -0.0192   -0.1228
   -0.0862    0.0264   -0.0684   -0.0769    0.0226   -0.1190   -0.1128
   -0.0830    0.0368   -0.0591   -0.1632   -0.1173   -0.1522   -0.1189
   -0.0859    0.0244   -0.0389   -0.1548   -0.0878   -0.1780   -0.1010
   -0.0847    0.0179   -0.0365   -0.1492   -0.0782   -0.2174   -0.0783
   -0.0848    0.0213   -0.0990    0.0079   -0.1512   -0.1206   -0.2159
   -0.0848    0.0531   -0.1299    0.0658   -0.1122   -0.1350   -0.1465
   -0.0823    0.0470   -0.1088    0.0433   -0.1327   -0.0054   -0.1284
   -0.0794    0.0452   -0.1036    0.0041   -0.0381   -0.0208   -0.1362
   -0.0793    0.0081   -0.1049    0.0109   -0.0191   -0.1558   -0.1004
   -0.0772   -0.1129   -0.1082   -0.1466   -0.1893   -0.0672   -0.1956
   -0.0771   -0.1054   -0.0853   -0.1263   -0.1561   -0.0846   -0.1585
   -0.0685   -0.1025   -0.0516   -0.1161   -0.0974   -0.0976   -0.0758
   -0.0682   -0.1037   -0.0497   -0.1395   -0.0617   -0.1156   -0.2307
   -0.0490    0.0153    0.0976    0.1095    0.2209   -0.1859    0.1773
   -0.0491    0.0635    0.0775    0.2661    0.1560   -0.2379    0.2458
   -0.0491    0.0735    0.1008    0.2393    0.1643   -0.3414    0.2162
   -0.0484    0.0823    0.0256    0.2703   -0.2947    0.3720   -0.5297
   -0.0468    0.0810   -0.0497    0.4337   -0.2108    0.6123   -0.1247
   -0.0476    0.0832    0.0045    0.4697   -0.3135   -0.1630   -0.4247
   -0.0458    0.1785    0.0962   -0.2320   -0.1288   -0.4377   -0.3663
   -0.0169    0.1461    0.1856   -0.2225   -0.1471   -0.4298   -0.3595
   -0.0160    0.1546    0.2601   -0.3392   -0.1774   -0.6324   -0.3104
   -0.0200    0.1692   -0.1904    0.6476   -0.0953    1.5090   -0.3750
   -0.0188   -0.1585    0.1591   -0.3765   -0.1083   -0.5855   -0.1644
   -0.0166   -0.1610    0.1795   -0.2876   -0.1199   -0.4391   -0.9631
   -0.0826    0.1465   -0.0245    0.5829   -0.2227    1.3258   -1.0776
   -0.0822    0.0837   -0.0136    0.3037    0.1084   -0.3700   -0.1171
   -0.0835    0.0951   -0.0802    0.3142   -0.0817   -0.2554   -0.2916
   -0.0835    0.1001   -0.0594    0.3198   -0.0727   -0.2914   -0.2599
   -0.0829    0.0980   -0.0466    0.3372   -0.0404   -0.2813   -0.3146
   -0.0855    0.0499   -0.0685    0.2441   -0.0948   -0.2515   -0.1439
   -0.0855    0.0613   -0.0317    0.2769   -0.2129    0.4561   -1.0670
   -0.0888    0.0562   -0.0877    0.2220   -0.1163    0.4775   -0.8526
   -0.0875    0.0821   -0.0031    0.2910   -0.1327    0.5339   -0.1144
   -0.0844    0.0805   -0.0156    0.2852   -0.1308    0.5425   -0.0681
   -0.0816    0.1058   -0.0766    0.3652   -0.1660    0.4796   -0.1436
   -0.0813    0.1039   -0.0767    0.3493   -0.1633    0.4632   -0.1379
   -0.0813    0.1081   -0.1089   -0.1026   -0.1603   -0.2542   -0.6089
   -0.0813    0.0356   -0.2427   -0.1728   -0.1198   -0.3202   -0.9485
   -0.0994   -0.0949   -0.1010   -0.1010   -0.1224   -0.2479   -0.8680
   -0.0993   -0.0950   -0.1000   -0.0904   -0.1244   -0.2282   -0.8458
   -0.0993   -0.0950   -0.0992   -0.1232   -0.0944   -0.1112   -0.6426
   -0.0995   -0.0950   -0.0992   -0.1229   -0.2466   -0.0925   -0.8648
   -0.0996   -0.1002   -0.0999   -0.1225   -0.2465   -0.0917   -0.8637
   -0.0994   -0.0999   -0.0991   -0.1232   -0.2410   -0.0930   -0.8562
   -0.0994   -0.1000   -0.1185   -0.2230   -0.1375   -0.2867   -0.9426

  Columns 8 through 12

    0.1778   -0.3011   -0.0711   -0.2036   -0.0914
    0.4174    0.0266   -0.1402   -0.0866   -0.1046
    0.1758    0.0512   -0.0583   -0.1196   -0.0911
   -0.0313   -0.0778   -0.0958   -0.0891   -0.1121
   -0.1999   -0.3035   -0.1878   -0.1771   -0.0456
   -0.0963   -0.0617   -0.0161   -0.1833   -0.0241
   -0.1188   -0.2976    0.0412   -0.8922    0.6755
   -0.2053   -0.2511   -0.3370   -0.0065   -1.0121
    0.5417    0.2136    1.0482    1.1134    1.1363
   -0.2010   -0.1179   -0.3220   -0.8586   -0.9048
   -0.2898   -0.1794   -0.5839   -1.8037   -2.0086
   -0.2293   -0.0893   -0.3250   -0.8867   -0.9276
    0.7056    0.2319    0.2937    0.2170    0.2595
    0.0786   -0.1030   -0.2526   -0.3295   -0.3585
   -0.1123   -0.2047   -0.5261   -0.7250   -0.7909
   -0.2189   -0.1394   -0.5723   -0.7802   -0.7822
   -0.2498   -0.1550   -0.3446   -0.9485   -1.0028
   -0.2628   -0.1184   -0.5926   -0.6444   -0.6586
   -0.1966   -0.0867   -0.4981   -0.5401   -0.5374
   -0.1285   -0.5592   -0.5755   -0.7610   -0.7431
   -0.1366   -0.8247   -1.6045   -1.6353   -1.5878
   -0.0033   -0.7740   -1.2633   -1.2628   -1.1696
   -0.0309   -0.8590   -1.4114   -1.4109   -1.3027
   -0.2054   -1.0622   -1.6227   -1.6180   -1.6175
   -0.0283   -0.9161   -0.9168   -0.9167   -0.9167
   -0.0563   -0.8866   -0.8872   -0.8871   -0.8871
   -0.0925   -0.7762   -0.7768   -0.7767   -0.7767
   -0.9243   -2.3047   -2.4425   -2.4501   -2.4511
   -0.5156   -1.2340   -1.3052   -1.3091   -1.3096
   -0.4719   -0.9973   -1.9685   -1.9654   -1.9649
   -0.6906   -1.3573   -1.5013   -1.5034   -1.5034
   -0.5665   -1.0128   -0.9920   -0.9918   -0.9917
    0.4390    0.5210    0.6924    0.6760    0.6761
   -1.0730   -1.1766   -1.4640   -1.4350   -1.4350
   -1.4196   -1.8448   -2.0287   -1.9665   -1.9644
   -1.3892   -1.8037   -1.9831   -1.9224   -1.9203
   -1.8840   -2.4588   -2.7038   -2.6196   -2.6167
    1.2118    1.6361    1.6048    1.6053    1.6052
   -1.5954   -1.8403   -1.7917   -1.7906   -1.7903
   -1.2192   -1.3287   -1.2884   -1.2870   -1.2867
    0.0702    0.1934    0.1930    0.1936    0.1936
   -1.2033   -1.1975   -1.1993   -1.1993   -1.1993
   -1.2968   -1.2860   -1.2874   -1.2874   -1.2874
   -1.3060   -1.2952   -1.2967   -1.2967   -1.2967
   -0.8965   -0.9000   -0.9003   -0.9003   -0.9003
   -0.8154   -0.8185   -0.8187   -0.8187   -0.8187
   -0.1084   -0.1038   -0.1035   -0.1035   -0.1034
    0.0962    0.1008    0.1011    0.1012    0.1012
   -0.1322   -0.1339   -0.1340   -0.1340   -0.1340
   -0.0859   -0.0875   -0.0877   -0.0877   -0.0877
   -0.1869   -0.1911   -0.1915   -0.1915   -0.1915
   -0.1539   -0.1554   -0.1555   -0.1555   -0.1555
   -0.6079   -0.6078   -0.6078   -0.6078   -0.6078
   -0.7418   -0.7215   -0.7196   -0.7194   -0.7194
   -0.6853   -0.6673   -0.6656   -0.6654   -0.6654
   -0.6696   -0.6523   -0.6506   -0.6504   -0.6504
   -0.5214   -0.5095   -0.5083   -0.5082   -0.5082
   -0.6922   -0.6752   -0.6736   -0.6734   -0.6734
   -0.6914   -0.6744   -0.6728   -0.6726   -0.6726
   -0.6856   -0.6688   -0.6672   -0.6670   -0.6670
   -0.7403   -0.7204   -0.7185   -0.7183   -0.7183


cumulativeprofit6 =

   1.0e+10 *

  Columns 1 through 7

    0.0004    0.0046    0.0045    0.0098   -0.0118    0.0222   -0.0593
    0.0013    0.0015    0.0103    0.0099    0.0436   -0.0164    0.0035
    0.0013    0.0025    0.0142    0.0377    0.0976   -0.1776    0.4750
   -0.0030    0.0033    0.0168    0.0697   -0.0648    0.0736   -0.2982
   -0.0003   -0.0136    0.0231   -0.0293    0.0647    0.0142    0.0078
   -0.0083    0.0073   -0.0173    0.0321   -0.0442    0.0485   -0.2156
   -0.0082    0.0126   -0.0238    0.0792   -0.0572    0.2289   -0.5372
   -0.0077    0.0134   -0.0255   -0.0248   -0.0052   -0.0554    0.1079
```

```
-0.0075   0.0132  -0.0437   0.0001  -0.0772   0.1422  -0.4501
-0.0075   0.0122  -0.0380   0.0782  -0.1056   0.3887  -0.9413
-0.0071   0.0126  -0.0364   0.0582  -0.1114   0.1117  -0.0351
-0.0075   0.0132  -0.0261   0.0691  -0.1051   0.2136   0.0732
-0.0080   0.0102  -0.0337   0.0449  -0.0736   0.0394  -0.0339
-0.0080   0.0101  -0.0268   0.0382  -0.0383   0.0357  -0.0164
-0.0079  -0.0114  -0.0036  -0.0185  -0.0012  -0.0348  -0.0646
-0.0088  -0.0103  -0.0069  -0.0125  -0.0049  -0.0189  -0.0428
-0.0088  -0.0101  -0.0069  -0.0123  -0.0049  -0.0184  -0.0546
-0.0090  -0.0103  -0.0071  -0.0124  -0.0059  -0.0172  -0.0547
-0.0090  -0.0103  -0.0068  -0.0123  -0.0057  -0.0132  -0.0522
-0.0090  -0.0103  -0.0076  -0.0110  -0.0074  -0.0153  -0.0538
-0.0073  -0.0112  -0.0055  -0.0115  -0.0044  -0.0145  -0.0507
-0.0071  -0.0116  -0.0054  -0.0110  -0.0055  -0.0144  -0.0514
-0.0076  -0.0104  -0.0063  -0.0095  -0.0065  -0.0104  -0.0496
-0.0076  -0.0103  -0.0065  -0.0100  -0.0071  -0.0115  -0.0512
-0.0075  -0.0102  -0.0069  -0.0098  -0.0078  -0.0116  -0.0515
-0.0079  -0.0095  -0.0066  -0.0089  -0.0082  -0.0105  -0.0505
-0.0078  -0.0095  -0.0070  -0.0096  -0.0085  -0.0111  -0.0514
-0.0076   0.0001  -0.0083  -0.0012  -0.0064  -0.0067  -0.0435
-0.0076   0.0006  -0.0083  -0.0007  -0.0060  -0.0038  -0.0386
-0.0077  -0.0001  -0.0092  -0.0013  -0.0067  -0.0036  -0.0341
-0.0077  -0.0088  -0.0071  -0.0089  -0.0078  -0.0080  -0.0440
-0.0077  -0.0088  -0.0071  -0.0087  -0.0079  -0.0090  -0.0453
-0.0077  -0.0088  -0.0071  -0.0099  -0.0100  -0.0112  -0.0509
-0.0074  -0.0099  -0.0075  -0.0089  -0.0071  -0.0093  -0.0457
-0.0073  -0.0101  -0.0061  -0.0097  -0.0059  -0.0079  -0.0441
-0.0073  -0.0104  -0.0060  -0.0097  -0.0050   0.0003   0.0018
-0.0073  -0.0104  -0.0090  -0.0099  -0.0086  -0.0072  -0.0113
-0.0085  -0.0005  -0.0116  -0.0110  -0.0172  -0.0239  -0.0398
-0.0085  -0.0001  -0.0097   0.0030   0.0157   0.0326   0.0504
-0.0085  -0.0001  -0.0097   0.0030   0.0157   0.0326  -0.0372
-0.0087  -0.0021  -0.0143   0.0005   0.0111   0.0257  -0.0325
-0.0087  -0.0021  -0.0143   0.0005   0.0111  -0.0243   0.0488
-0.0091  -0.0021  -0.0157   0.0010   0.0125  -0.0253   0.0499
-0.0091  -0.0021  -0.0157   0.0010   0.0125   0.0282  -0.0354
-0.0089   0.0002  -0.0093   0.0033   0.0161   0.0332  -0.0386
-0.0087   0.0020  -0.0128   0.0053   0.0203   0.0408  -0.0437
-0.0087  -0.0001  -0.0097   0.0035   0.0084   0.0353  -0.0410
-0.0087  -0.0001  -0.0097   0.0035   0.0168   0.0305  -0.0373
-0.0087  -0.0031  -0.0106  -0.0008   0.0028   0.0117  -0.0247
-0.0087  -0.0031  -0.0110  -0.0010   0.0090   0.0231  -0.0325
-0.0087  -0.0031  -0.0110  -0.0010  -0.0091   0.0024  -0.0338
-0.0087  -0.0013  -0.0107  -0.0010  -0.0092  -0.0154  -0.0527
-0.0086  -0.0009  -0.0120   0.0041  -0.0116  -0.0104  -0.0469
-0.0084   0.0006  -0.0078  -0.0097  -0.0106  -0.0222  -0.0588
-0.0084   0.0006  -0.0078  -0.0114  -0.0161  -0.0227  -0.0599
-0.0084   0.0006  -0.0096  -0.0101  -0.0097  -0.0096  -0.0459
-0.0085  -0.0008  -0.0178  -0.0121  -0.0182  -0.0274  -0.0647
-0.0101  -0.0108  -0.0101  -0.0100  -0.0101  -0.0101  -0.0465
-0.0101  -0.0108  -0.0101  -0.0100  -0.0101  -0.0101  -0.0465
-0.0101  -0.0108  -0.0101  -0.0100  -0.0101  -0.0101  -0.0465
-0.0101  -0.0108  -0.0101  -0.0100  -0.0101  -0.0465  -0.0467
```

```
  Columns 8 through 12

    0.1716   -0.4146    0.7900   -0.2038   -0.2333
   -0.0783    0.1219   -0.2987   -0.3041   -0.3024
   -1.1548    2.1574   -0.7247   -0.7042   -0.6900
    0.6146   -1.2705   -1.4544   -1.3843   -1.3828
    0.0371   -0.0425   -0.0760   -0.0733   -0.0731
    0.3890   -0.9290    0.5415   -1.6202   -1.9833
    1.1141   -2.4309   -2.2828   -3.2889   -3.1835
   -0.2395   -0.0206   -0.0653   -0.0640   -0.0641
   -0.0776   -0.1321   -0.1411   -0.1364   -0.1366
    0.0489    0.0011   -0.0048   -0.0053   -0.0053
   -0.0484   -0.0792   -0.0850   -0.0823   -0.0825
    0.0483    0.0177    0.0173    0.0175    0.0174
   -0.0411   -0.0769   -0.0825   -0.0827   -0.0827
   -0.0152   -0.0484   -0.0488   -0.0520   -0.0521
   -0.0677   -0.0907   -0.1423   -0.1538   -0.1541
   -0.0456   -0.0724   -0.1195   -0.1291   -0.1293
   -0.0559   -0.0816   -0.1307   -0.1318   -0.1417
   -0.0559   -0.0816   -0.1307   -0.1318   -0.1418
   -0.0533   -0.0795   -0.1281   -0.1291   -0.1388
   -0.0551   -0.0450   -0.0848   -0.0855   -0.0917
   -0.0519   -0.0425   -0.0817   -0.0824   -0.0883
   -0.0526   -0.0677   -0.0917   -0.0925   -0.0991
   -0.0506   -0.0650   -0.0893   -0.0901   -0.0966
   -0.0658   -0.0716   -0.0935   -0.1013   -0.1014
   -0.0663   -0.0720   -0.0987   -0.0992   -0.0992
   -0.0648   -0.0704   -0.0989   -0.1001   -0.1005
   -0.0660   -0.0717   -0.1000   -0.1012   -0.1016
   -0.0553   -0.0601   -0.0614   -0.0890   -0.0894
   -0.0487   -0.0527   -0.0539   -0.0823      -Inf
   -0.0426   -0.0762   -0.0765   -0.0765      -Inf
   -0.0442   -0.0442   -0.0442   -0.0442      -Inf
   -0.0455   -0.0455   -0.0456   -0.0456      -Inf
   -0.0512   -0.0512   -0.0512   -0.0512      -Inf
   -0.0460   -0.0460   -0.0460   -0.0460   -0.0460
   -0.0444   -0.0444   -0.0444   -0.0444   -0.0444
   -0.0340   -0.0342   -0.0342   -0.0342   -0.0342
   -0.0478   -0.0480   -0.0481   -0.0481   -0.0481
   -0.0779   -0.0782   -0.0782   -0.0782   -0.0782
    0.0173    0.0171    0.0171    0.0171    0.0171
    0.0021    0.0024    0.0024    0.0024    0.0024
   -0.0062   -0.0061   -0.0061   -0.0061   -0.0061
   -0.1139   -0.0081   -0.0075   -0.0074   -0.0074
   -0.1074   -0.0075   -0.0069   -0.0069   -0.0068
    0.0922   -0.0273   -0.0280   -0.0281   -0.0281
    0.1050   -0.0244   -0.0252   -0.0253   -0.0253
    0.1249   -0.0198   -0.0207   -0.0208   -0.0208
    0.1120   -0.0196   -0.0204   -0.0205   -0.0205
    0.0986   -0.1955    0.0280    0.0293    0.0294
    0.0485   -0.0340   -0.0346   -0.0346   -0.0346
   -0.0063   -0.0061   -0.0061   -0.0061   -0.0061
   -0.0340   -0.0341   -0.0341   -0.0341   -0.0341
   -0.0530   -0.0530   -0.0530   -0.0530   -0.0530
```

```
-0.0472   -0.0472   -0.0472   -0.0472   -0.0472
-0.0590   -0.0591   -0.0591   -0.0591   -0.0591
-0.0601   -0.0601   -0.0601   -0.0601   -0.0601
-0.0462   -0.0462   -0.0462   -0.0462   -0.0462
-0.0650   -0.0650   -0.0650   -0.0650   -0.0650
-0.0468   -0.0468   -0.0468   -0.0468   -0.0468
-0.0468   -0.0468   -0.0468   -0.0468   -0.0468
-0.0468   -0.0468   -0.0468   -0.0468   -0.0468
-0.0467   -0.0467   -0.0467   -0.0467   -0.0467
```





Published with MATLAB® R2018b