Testing numerical optimization of DSS and VSS using fme example

```
Adding the following folders to the path:
 -FTSC
 -Kalman
 -KPMstats
```

Contents

Clear

```
clear;
clc;

rng(1)                                       % control the randomness
```

******Testing for the fme example*********

```
n = 20;                                       % number of subjects
m = 30;                                       % number of observations
t = (1:m)/m;
p = 1;                                        % # of fixed effects
q = 1;                                        % # of random effects
sigma_e = 1;                                  % variance of white noise

d = 2*(p+n*q);                                % dimension of states

realFixedEffect = 7*sin(2*pi*t);              % p-by-m
realRandomEffect = randn(n,4)*[cos(2*pi*t);cos(4*pi*t);...
                               cos(6*pi*t);ones(1,m)];

Y = repmat(realFixedEffect, [n,1]) + realRandomEffect ...
    + sqrt(sigma_e)*randn(n,m);
```

Model setting

```
fixedDesign = repmat(ones(n,p),[1, 1, m]);    % n-by-p-by-m
randomDesign = repmat(ones(n,q),[1, 1, m]);   % n-by-q-by-m

%  Optimization
logpara0 = [0;                                      % log of e
         -10;-10;                                 % logs of lambdaF, lambdaR
         1*ones(2*q,1)];                         % log of randomDiag

diffusePrior = 1e7;

k = 1;
```

Optimization

```
DSS
```

```
NlogLik_dss = @(logpara) ...
    fme_dss_NlogLik(Y, fixedDesign, randomDesign, t, logpara, diffusePrior);

tic
[logparahat_dss, val_dss] = fminsearch(NlogLik_dss, logpara0);
toc
```

```
Elapsed time is 209.855631 seconds.
```

KF

```
NlogLik_vss = @(logpara) ...
    fme2KF(Y, fixedDesign, randomDesign, t, logpara, diffusePrior, true);

tic
[logparahat_vss, val_vss] = fminsearch(NlogLik_vss, logpara0);
toc
```

```
Elapsed time is 17.933446 seconds.
```

Model fitting

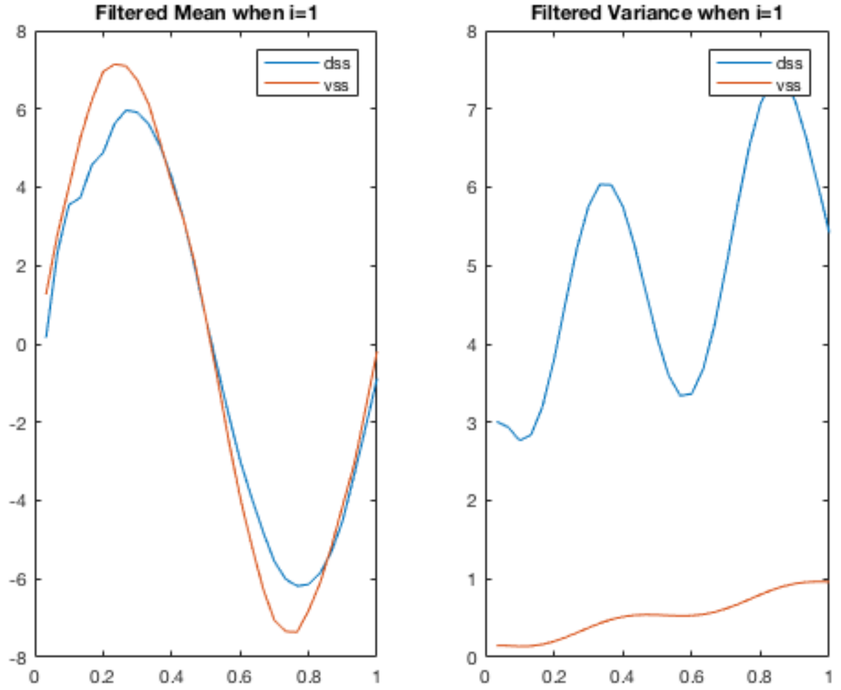
DSS

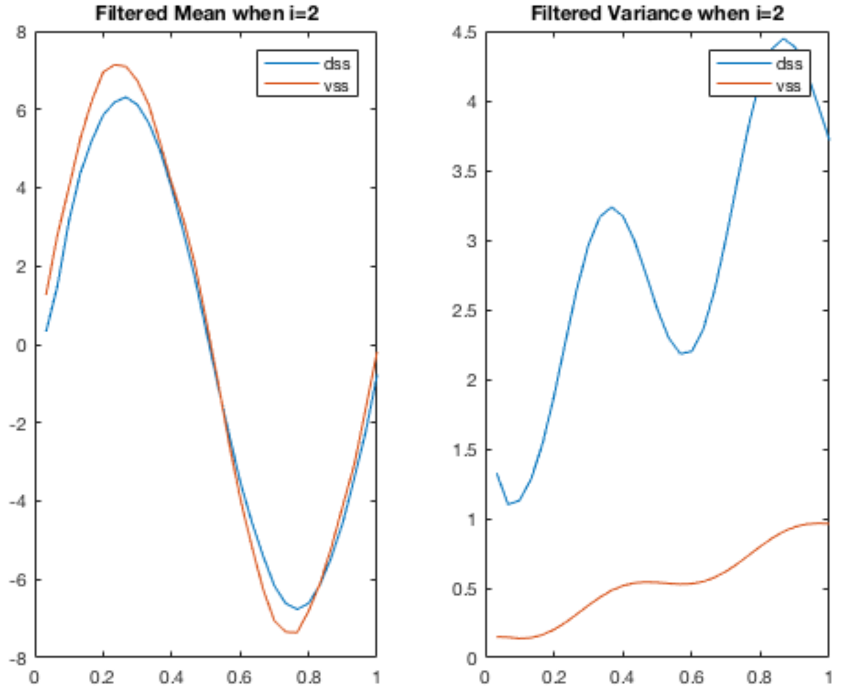
```
tic
[output_arg_dss, loglik, prior] = fme2dss(Y, fixedDesign, randomDesign, t, logparahat_dss, diffusePrior);
toc

%  KF
tic
output_arg_KF = fme2KF(Y, fixedDesign, randomDesign, t, logparahat_vss, diffusePrior, false);
toc

%  KS
tic
output_arg_KS = fme2KS(Y, fixedDesign, randomDesign, t, logparahat_vss, diffusePrior);
toc
```

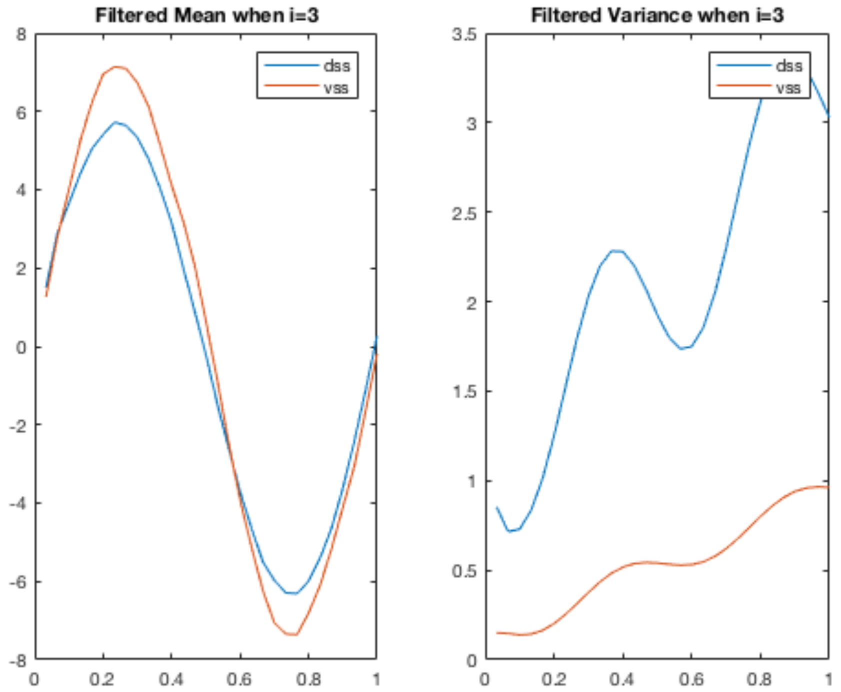
```
Elapsed time is 0.635903 seconds.
Elapsed time is 0.036000 seconds.
Elapsed time is 0.090313 seconds.
```

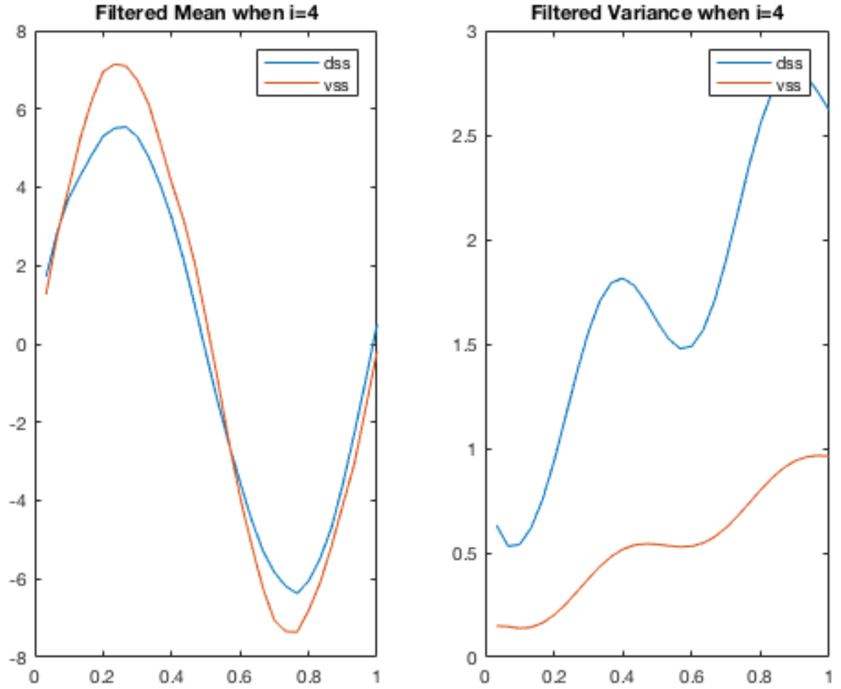
Filtering

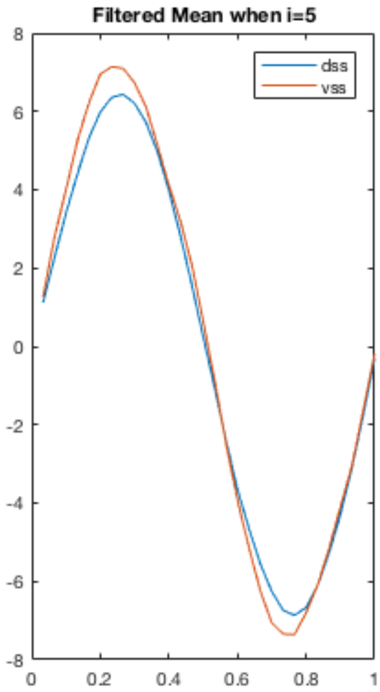
```
for i=1:n
    %  DSS
    fixedEffectMeanhat_dss = output_arg_dss{i}.FilteredMean(k,:);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% INSERT HERE
    fixedEffectCovhat_dss = reshape(output_arg_dss{i}.FilteredCov(k,k,:), [1, m]);

    %  KF
    fixedEffectMeanhat_KF = output_arg_KF.FilteredMean(k,:);
    fixedEffectCovhat_KF = reshape(output_arg_KF.FilteredCov(k,k,:), [1, m]);


    % Plotting
    figure;
    subplot(1,2,1)
    plot(t, fixedEffectMeanhat_dss, t, fixedEffectMeanhat_KF );
    legend("dss", "vss");
    plottitle = strcat("Filtered Mean when i=", num2str(i));
    title(plottitle);

    subplot(1,2,2)
    plot(t, fixedEffectCovhat_dss, t, fixedEffectCovhat_KF);
    legend("dss", "vss");
    plottitle = strcat("Filtered Variance when i=", num2str(i));
    title(plottitle);
end
```

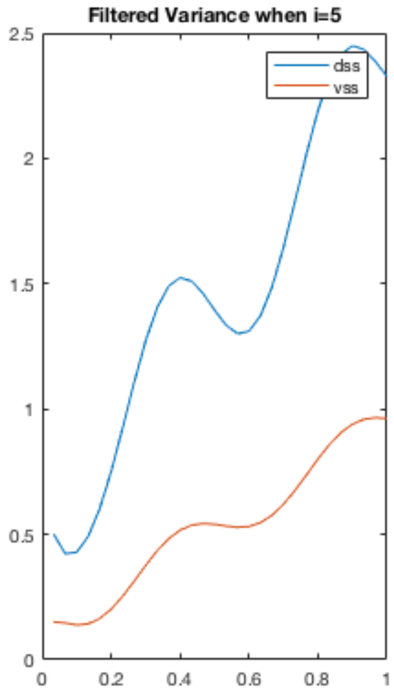


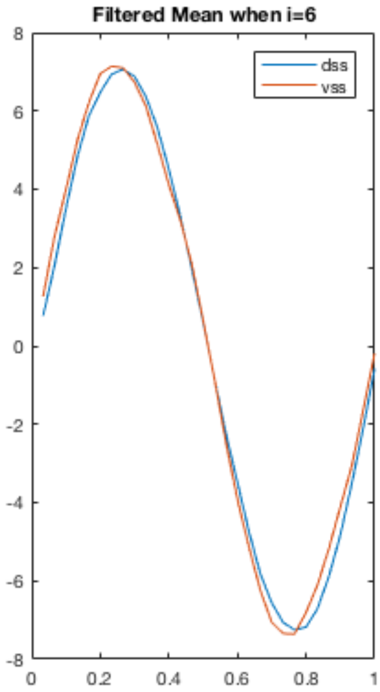


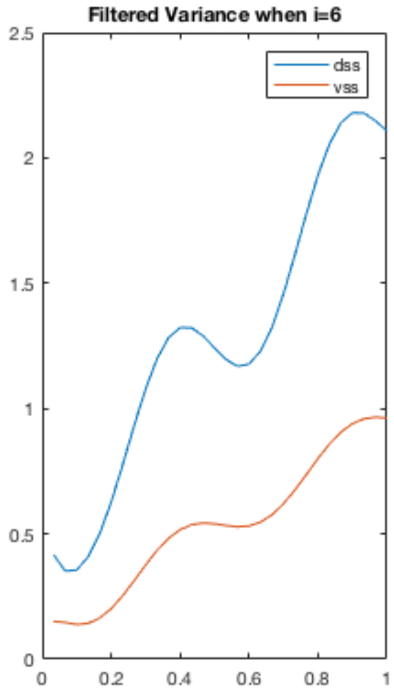

Filtered Mean when i=3
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Filtered Variance when i=3
dss
vss
3.5
3
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

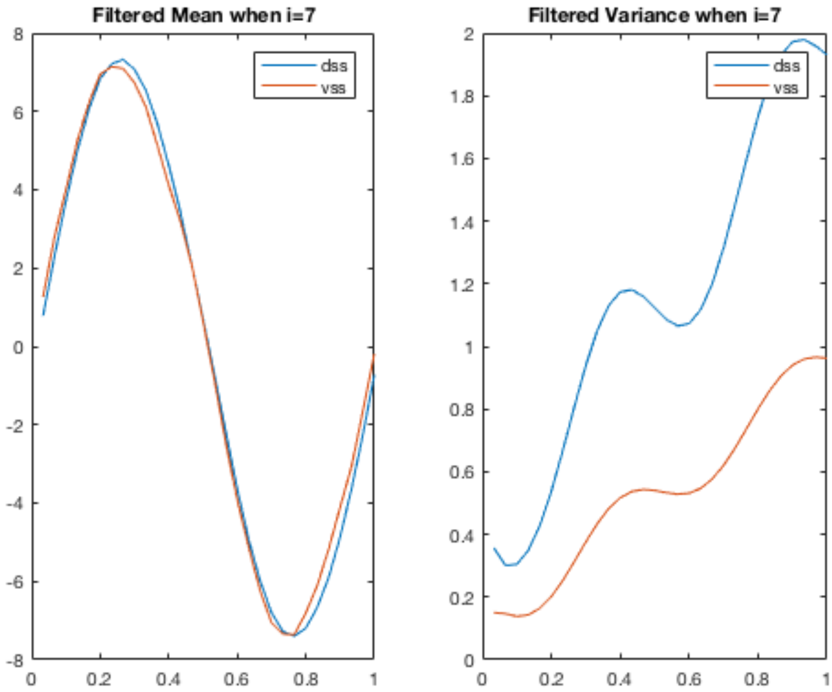

Filtered Mean when i=4
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Filtered Variance when i=4
dss
vss
3
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

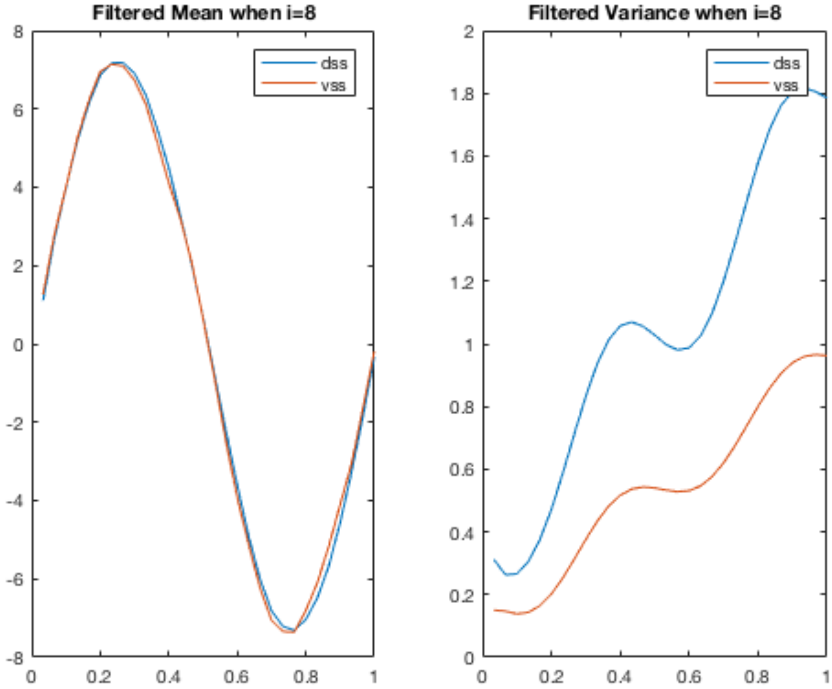

Filtered Mean when i=5
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1

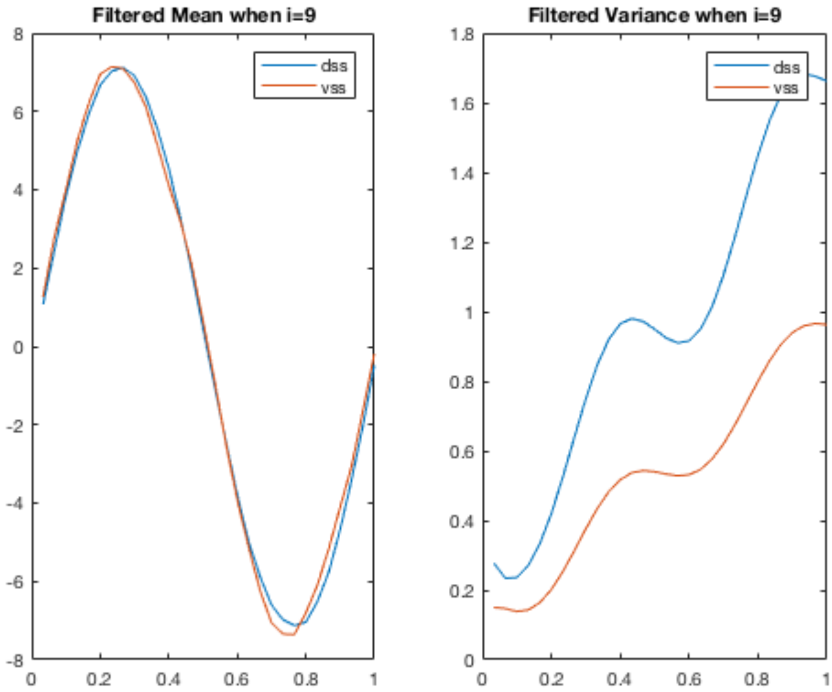

Filtered Variance when i=5
dss
vss
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

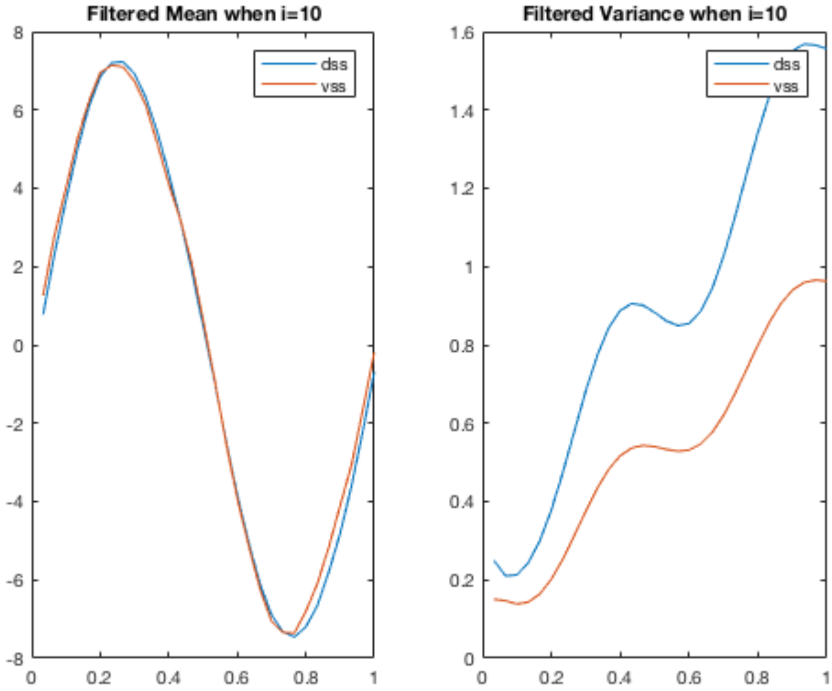

Filtered Mean when i=6
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1

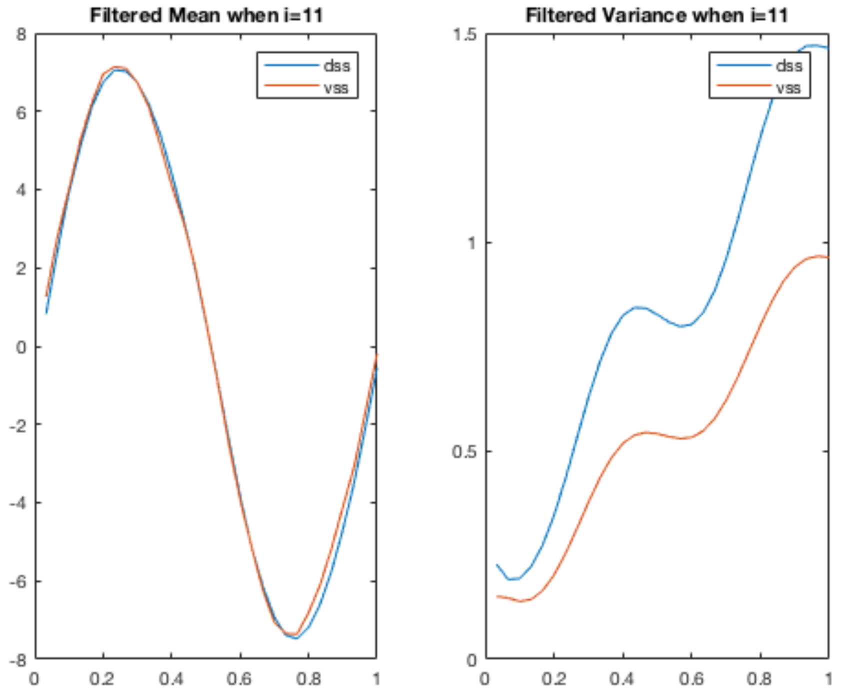

Filtered Variance when i=6
dss
vss
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

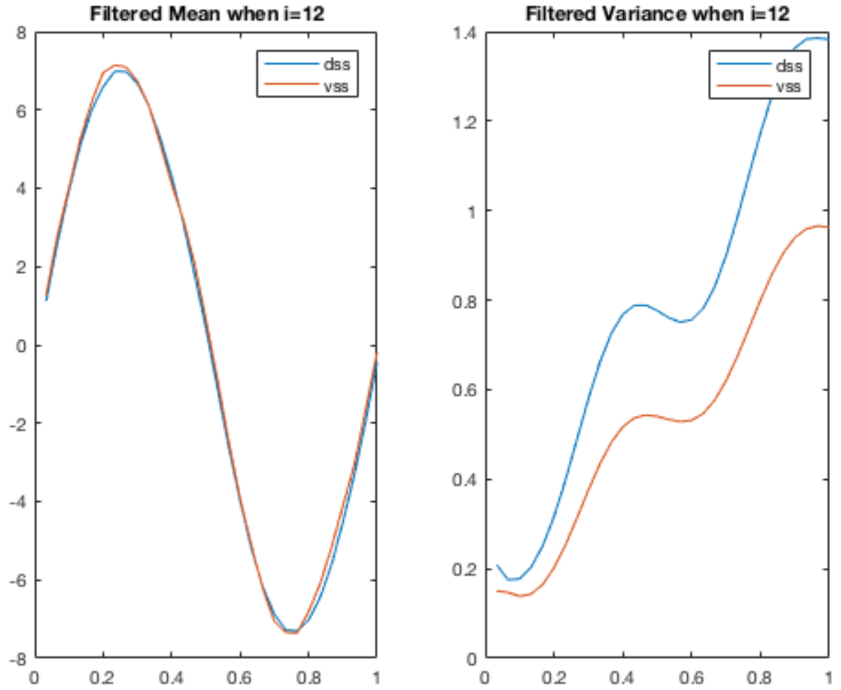

Filtered Mean when i=7
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Filtered Variance when i=7
dss
vss
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

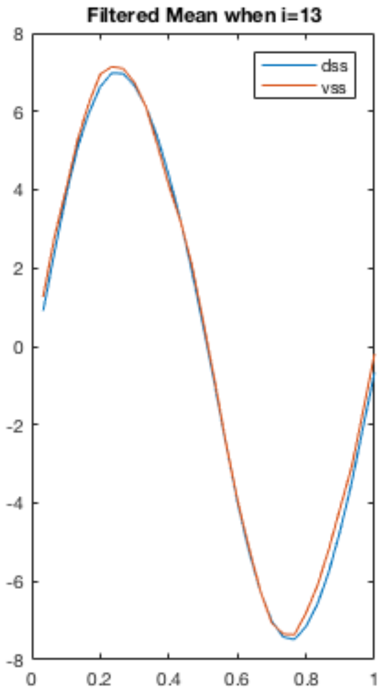

Filtered Mean when i=8
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Filtered Variance when i=8
dss
vss
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

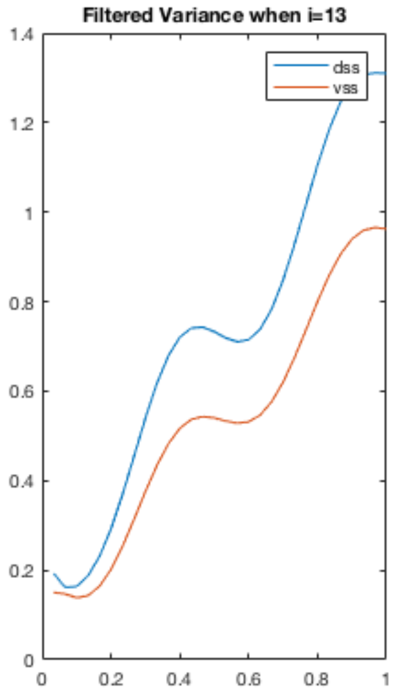

Filtered Mean when i=9
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Filtered Variance when i=9
dss
vss
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

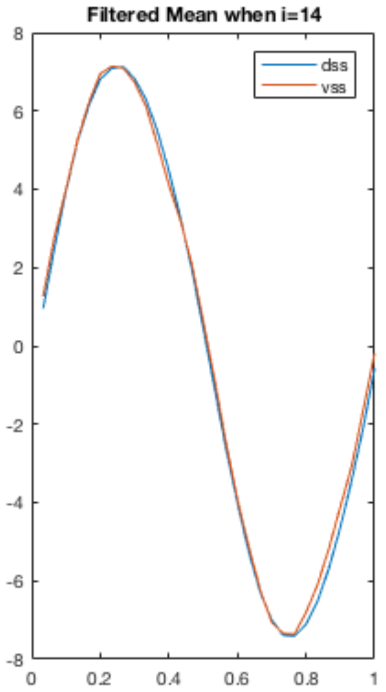

Filtered Mean when i=10
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Filtered Variance when i=10
dss
vss
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

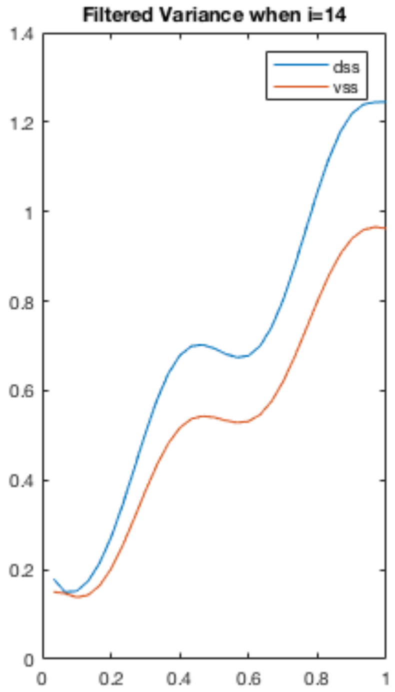

Filtered Mean when i=11
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Filtered Variance when i=11
dss
vss
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

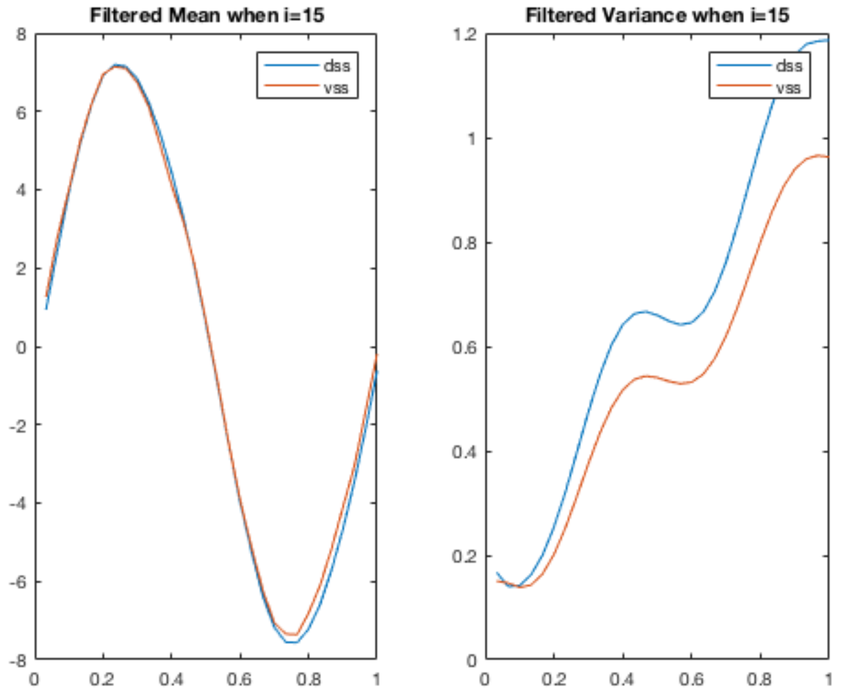

Filtered Mean when i=12
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Filtered Variance when i=12
dss
vss
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

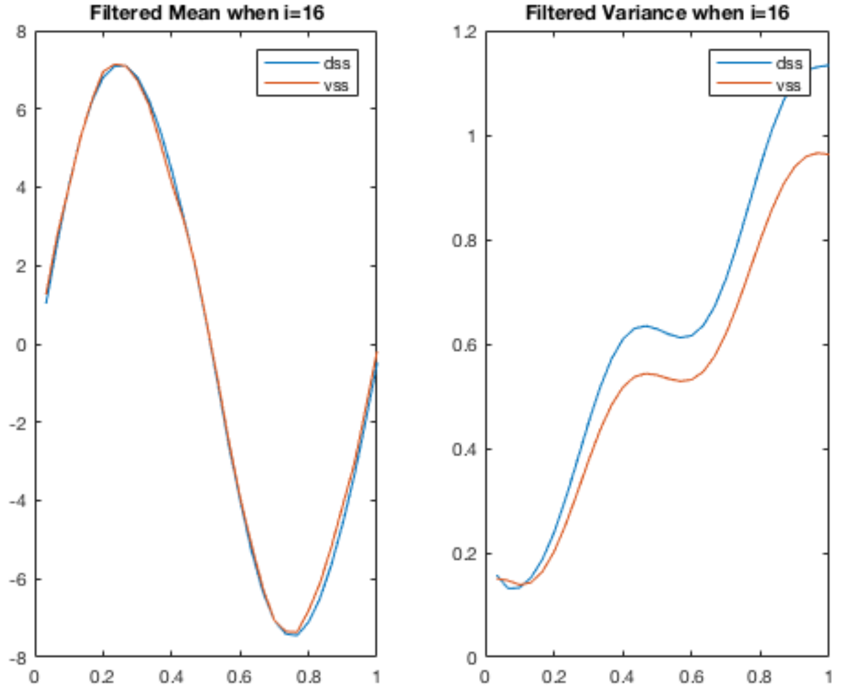

Filtered Mean when i=13

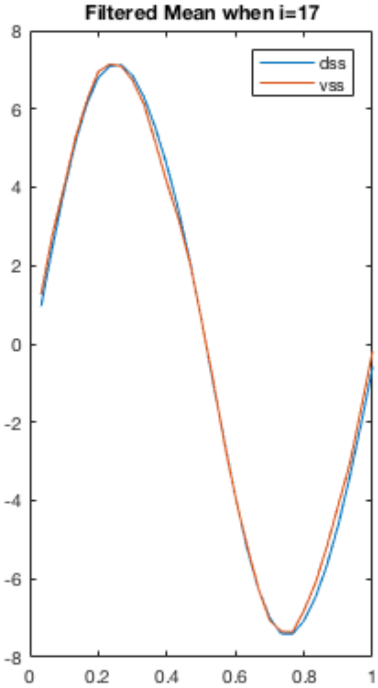

Filtered Variance when i=13

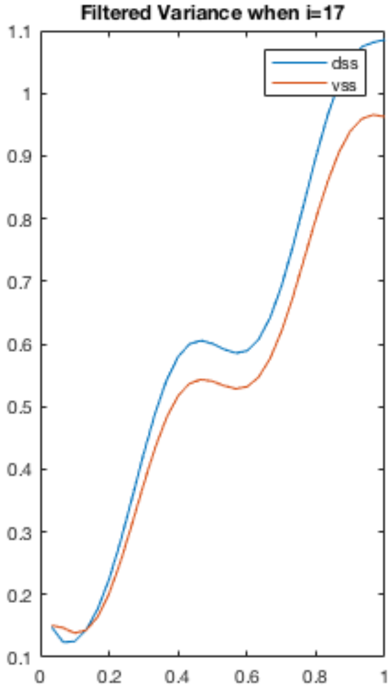

Filtered Mean when i=14

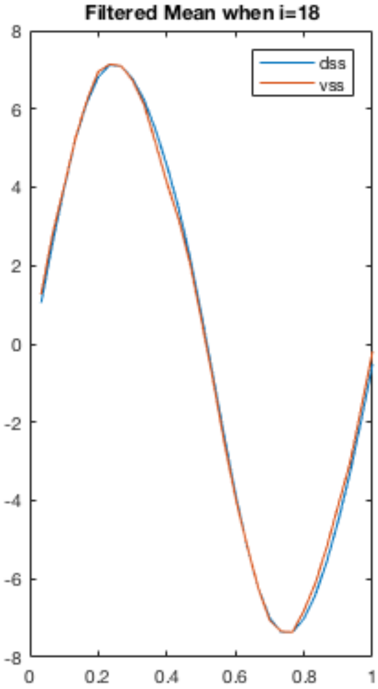

Filtered Variance when i=14

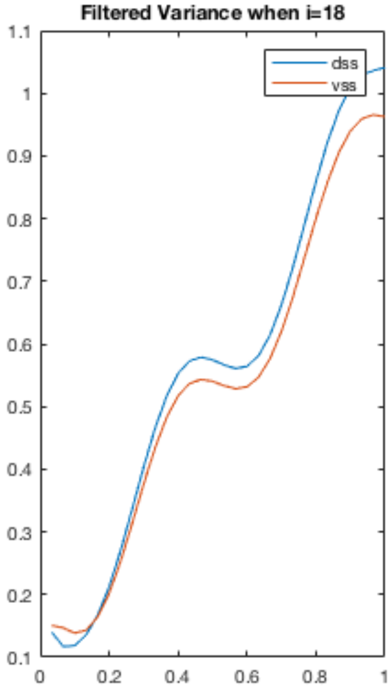

Filtered Mean when i=15
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Filtered Variance when i=15
dss
vss
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

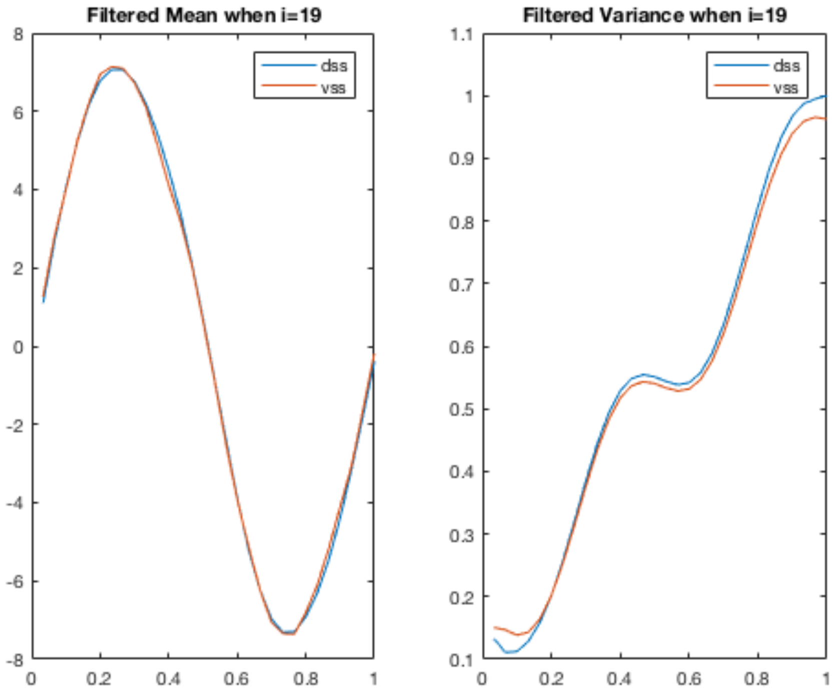

Filtered Mean when i=16
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Filtered Variance when i=16
dss
vss
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

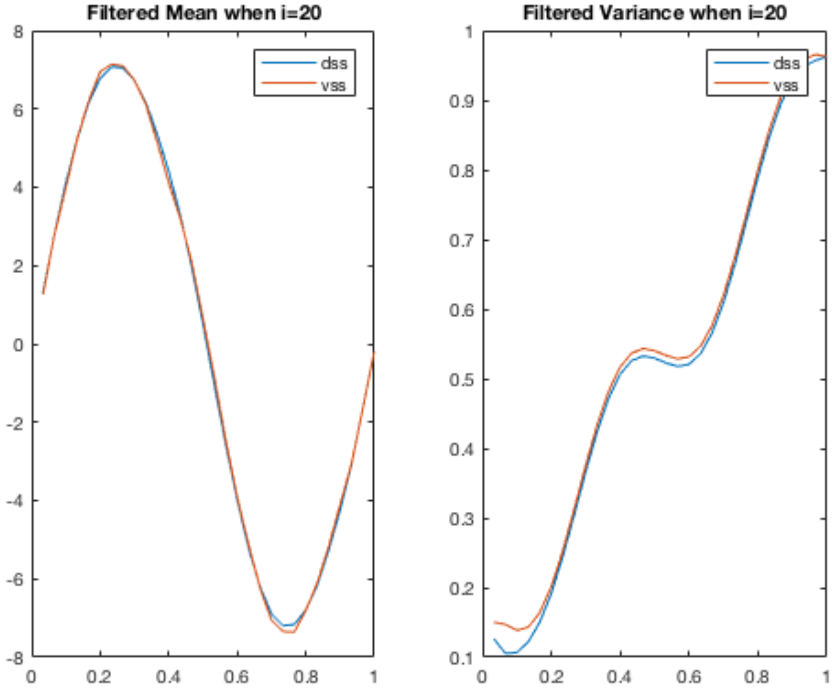

Filtered Mean when i=17
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1


Filtered Variance when i=17
dss
vss
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
1


Filtered Mean when i=18
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1

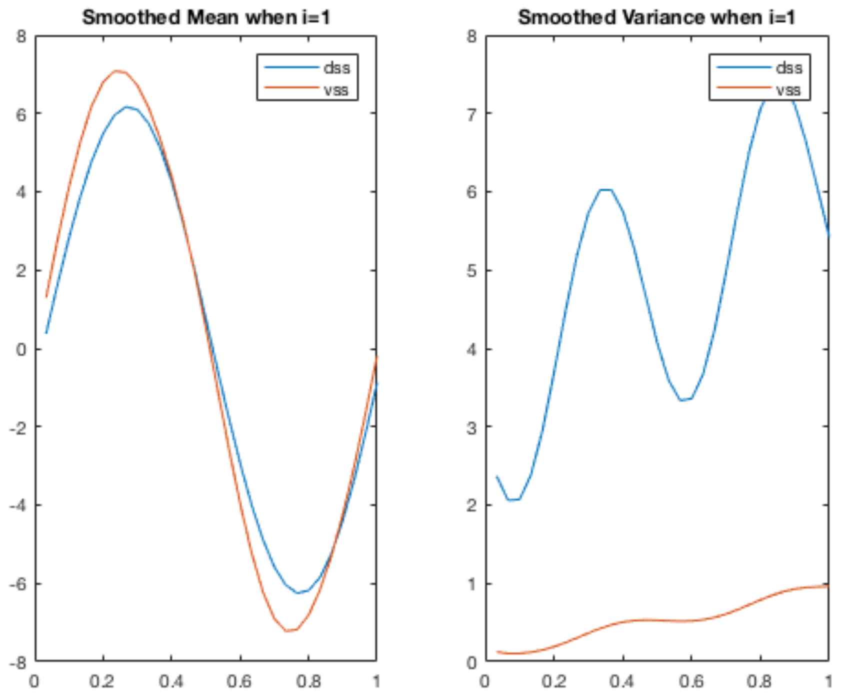

Filtered Variance when i=18
dss
vss
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
1

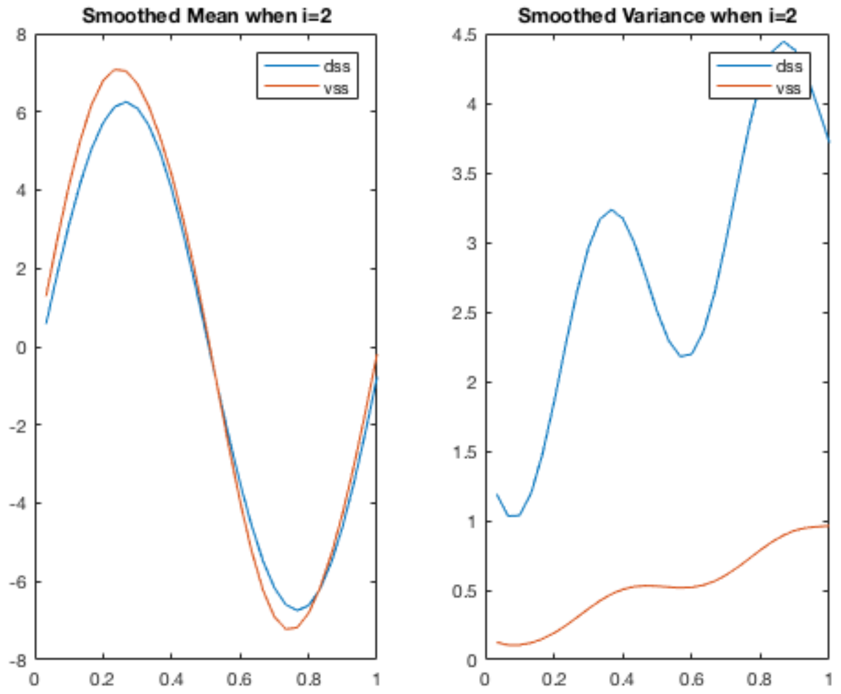


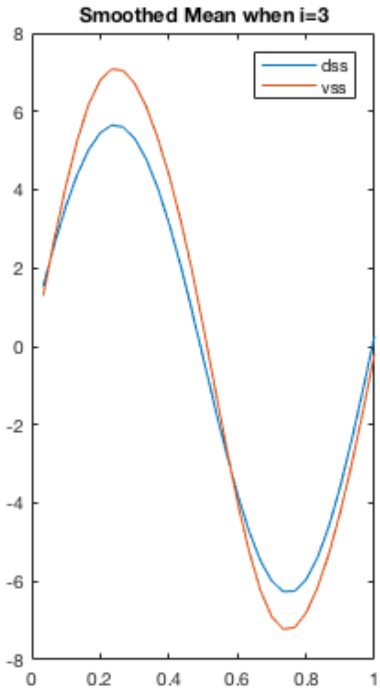


Smoothing

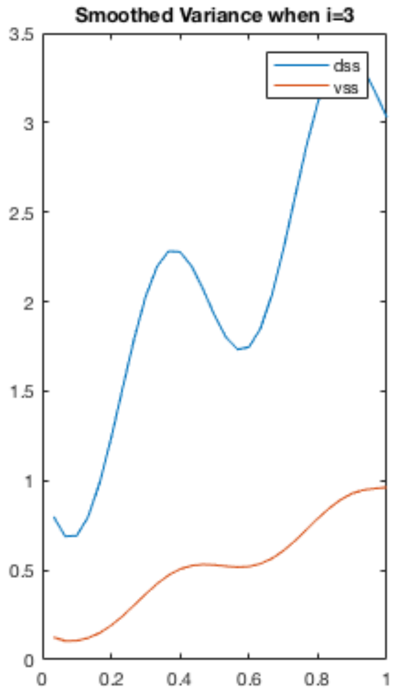
```
for i=1:n
    %  DSS
    fixedEffectMeanhat_dss = output_arg_dss{i}.SmoothedMean(k,:);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% INSERT HERE
    fixedEffectCovhat_dss = reshape(output_arg_dss{i}.SmoothedCov(k,k,:), [1, m]);

    %  KS
    fixedEffectMeanhat_KS = output_arg_KS.SmoothedMean(k, :);
    fixedEffectCovhat_KS = reshape(output_arg_KS.SmoothedCov(k,k,:), [1, m]);

    % Plotting
    figure;
    subplot(1,2,1)
    plot(t, fixedEffectMeanhat_dss, t, fixedEffectMeanhat_KS);
    legend("dss", "vss");
    plottitle = strcat("Smoothed Mean when i=", num2str(i));
    title(plottitle);

    subplot(1,2,2)
    plot(t, fixedEffectCovhat_dss, t, fixedEffectCovhat_KS);
    legend("dss", "vss");
    plottitle = strcat("Smoothed Variance when i=", num2str(i));
    title(plottitle);

end
```



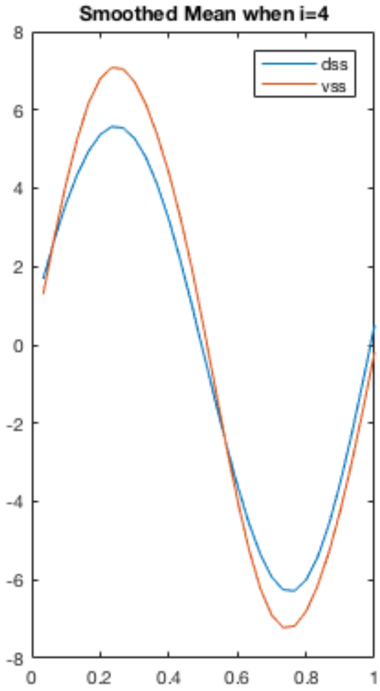




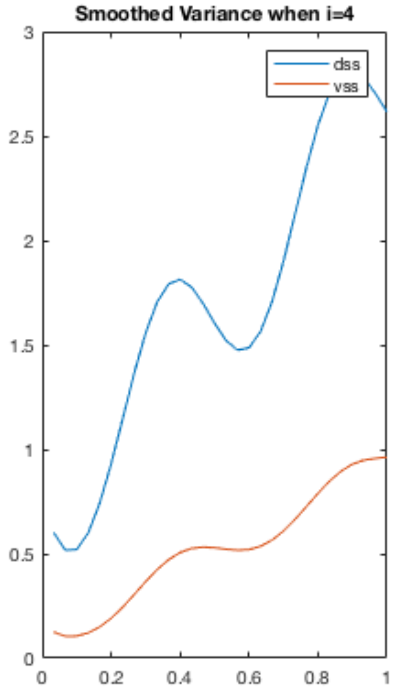
Smoothed Mean when i=3



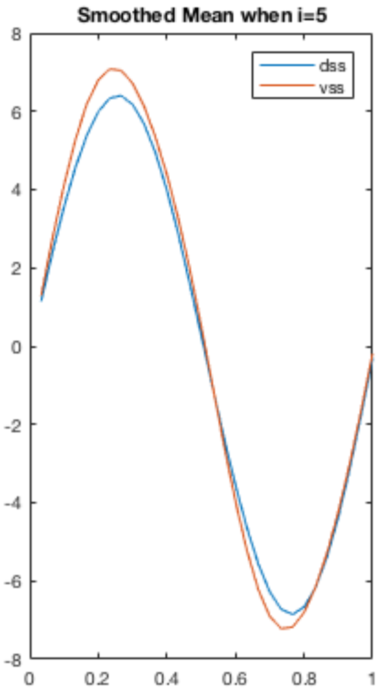
Smoothed Variance when i=3

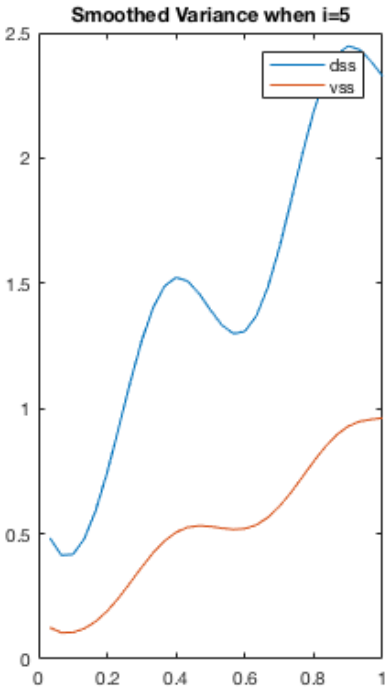


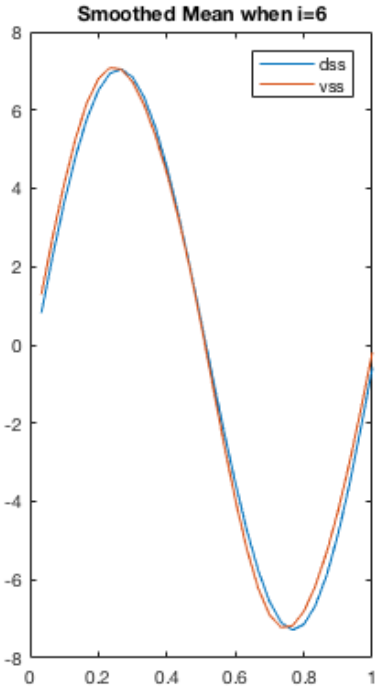
Smoothed Mean when i=4

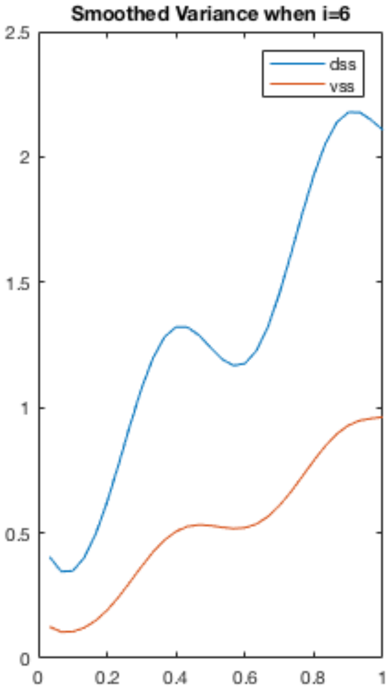


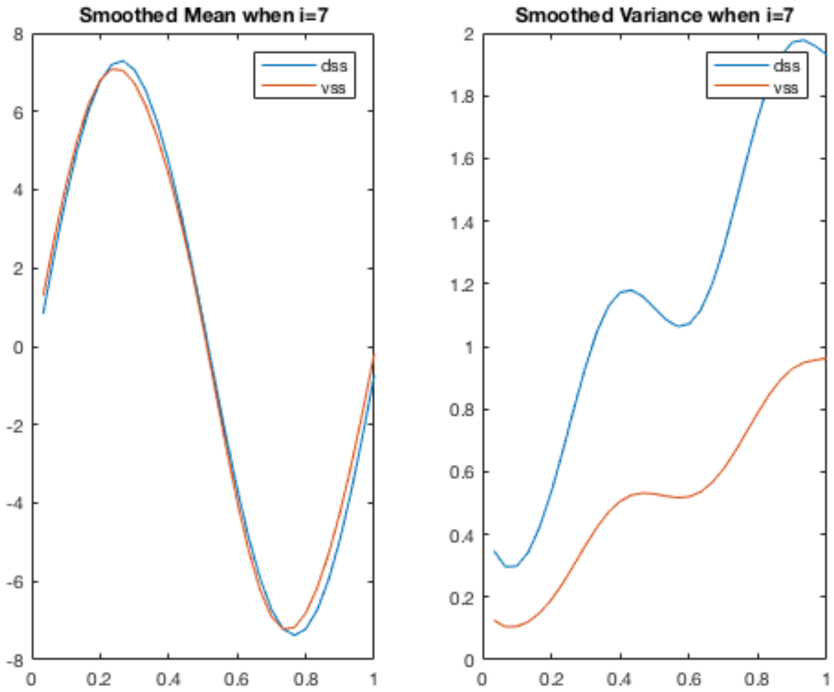
Smoothed Variance when i=4


Smoothed Mean when i=5
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1

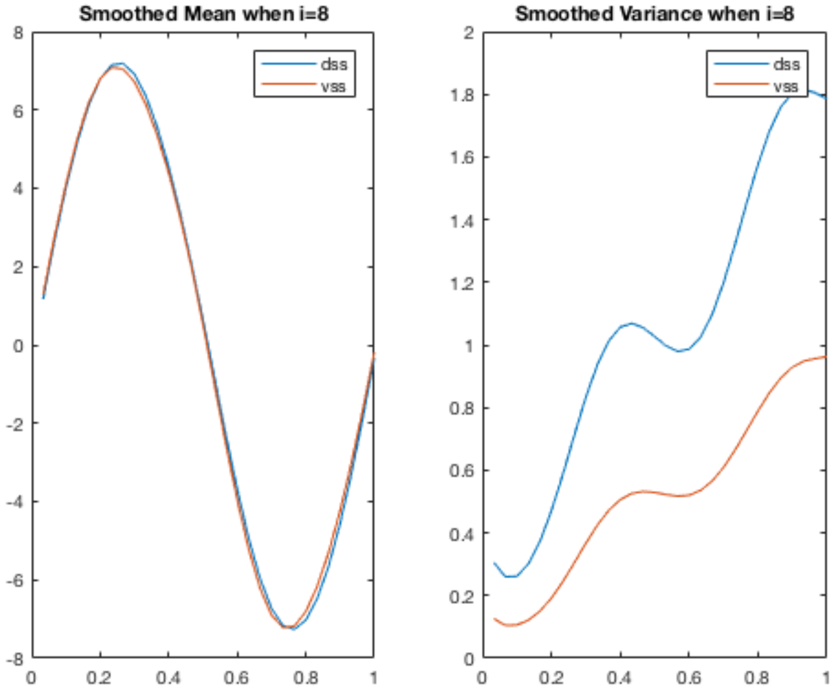

Smoothed Variance when i=5
dss
vss
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1


Smoothed Mean when i=6
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1

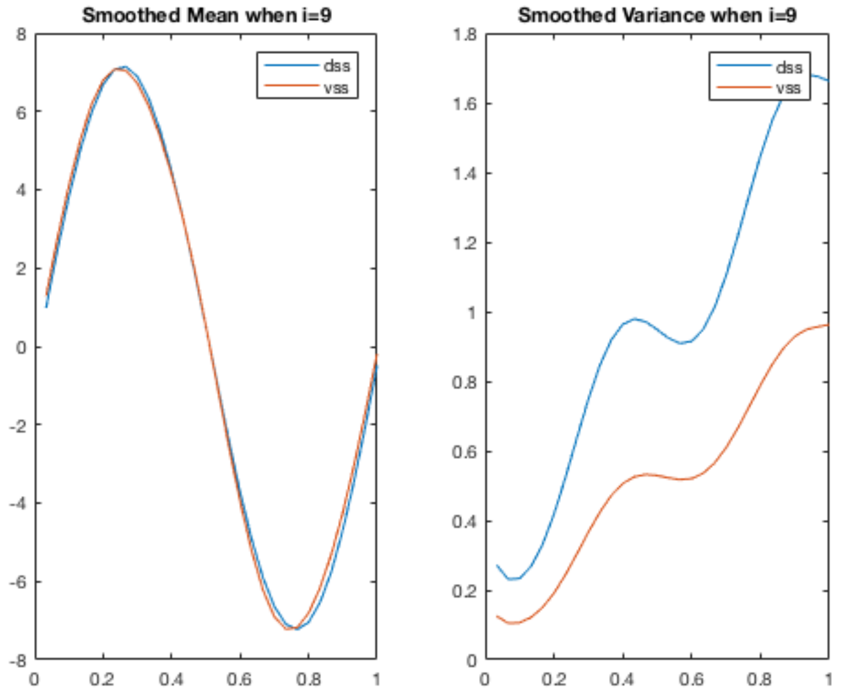

Smoothed Variance when i=6
dss
vss
2.5
2
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

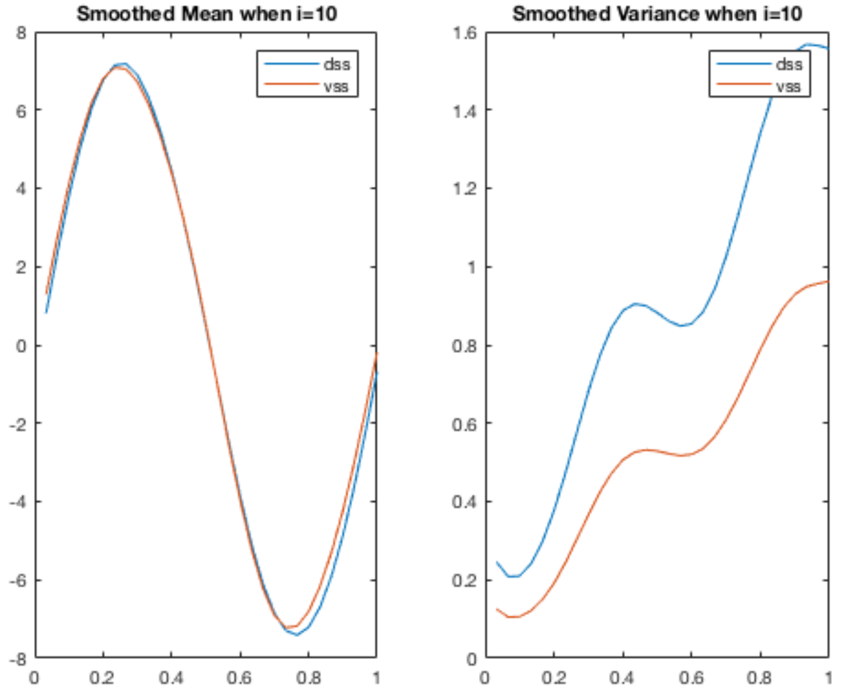

Smoothed Mean when i=7

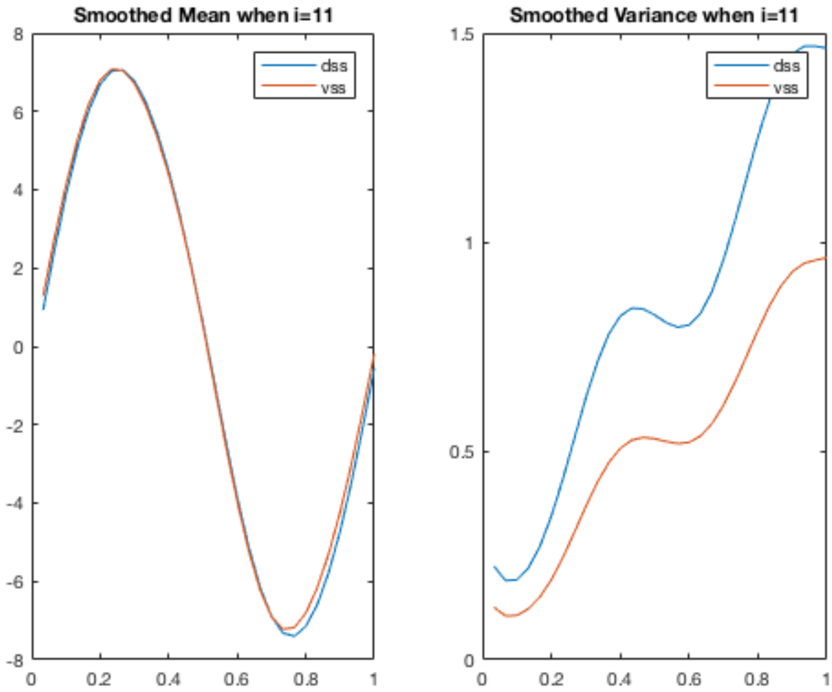
Smoothed Variance when i=7

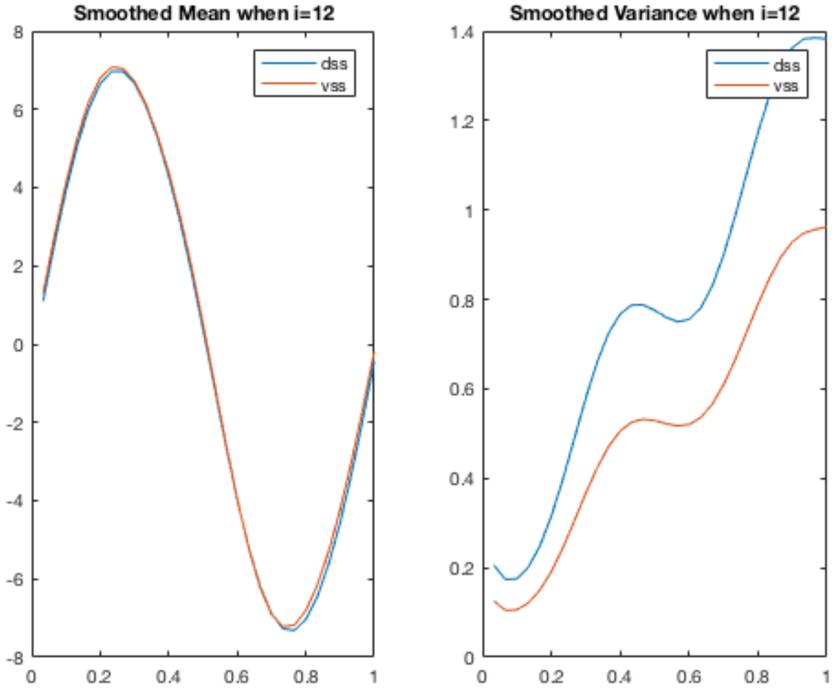

Smoothed Mean when i=8

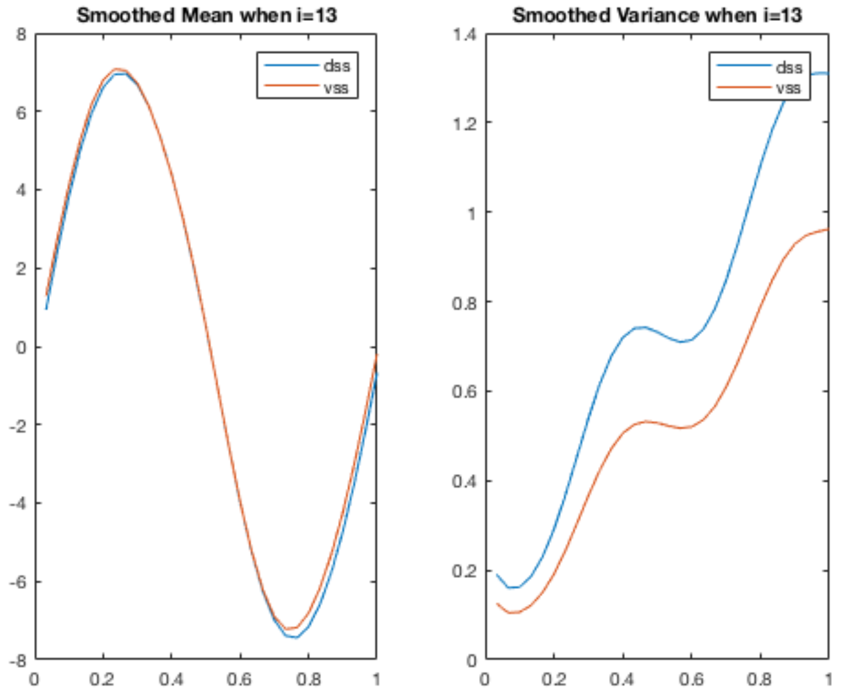
Smoothed Variance when i=8

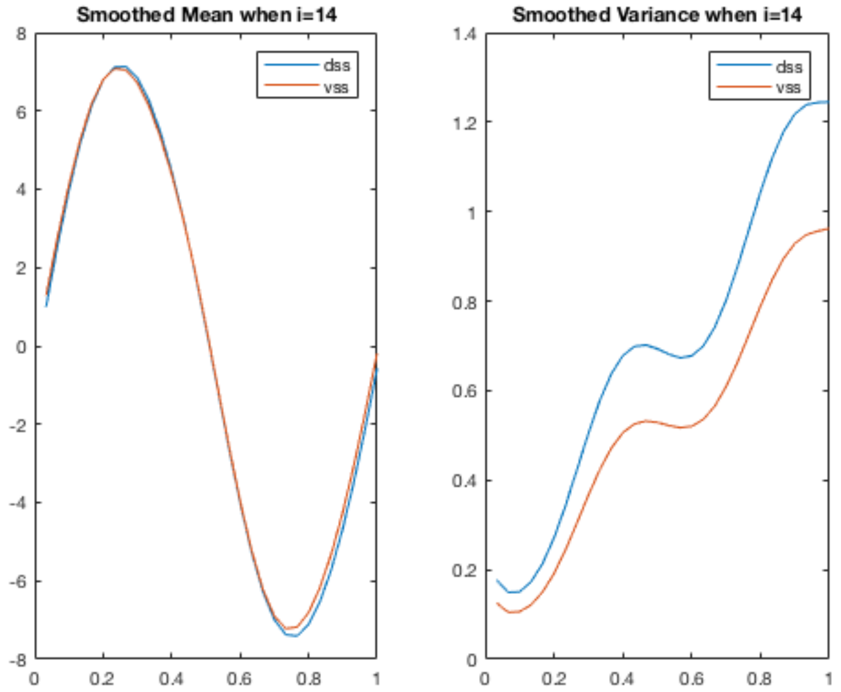

Smoothed Mean when i=9
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
dss
vss
Smoothed Variance when i=9
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
dss
vss

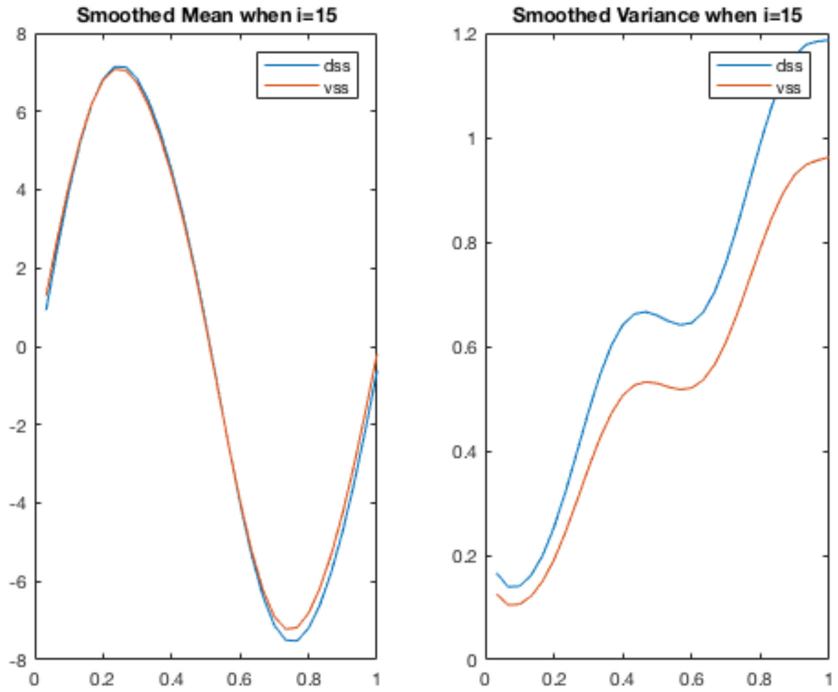

Smoothed Mean when i=10
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
dss
vss
Smoothed Variance when i=10
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
dss
vss

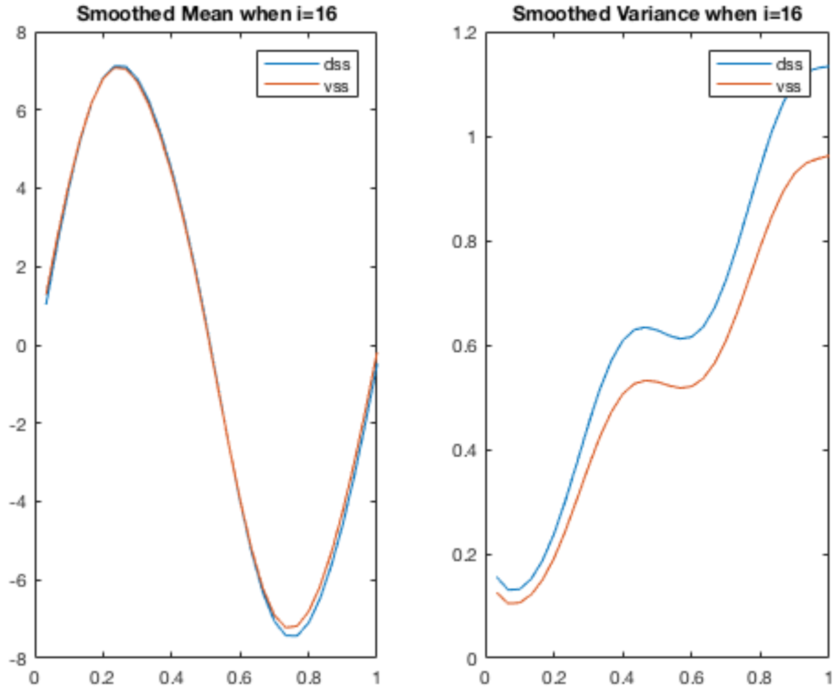

Smoothed Mean when i=11
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Smoothed Variance when i=11
dss
vss
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1

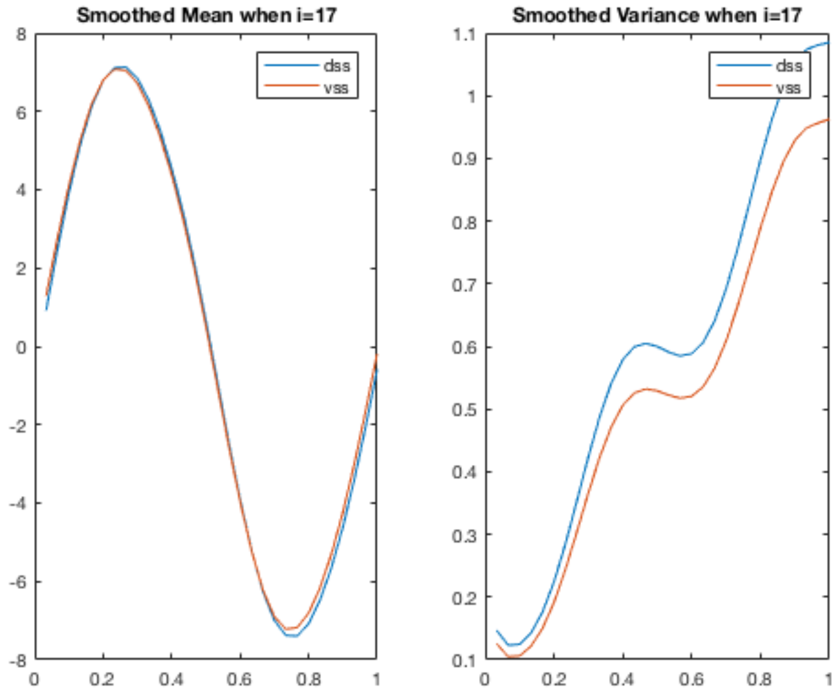

Smoothed Mean when i=12
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Smoothed Variance when i=12
dss
vss
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

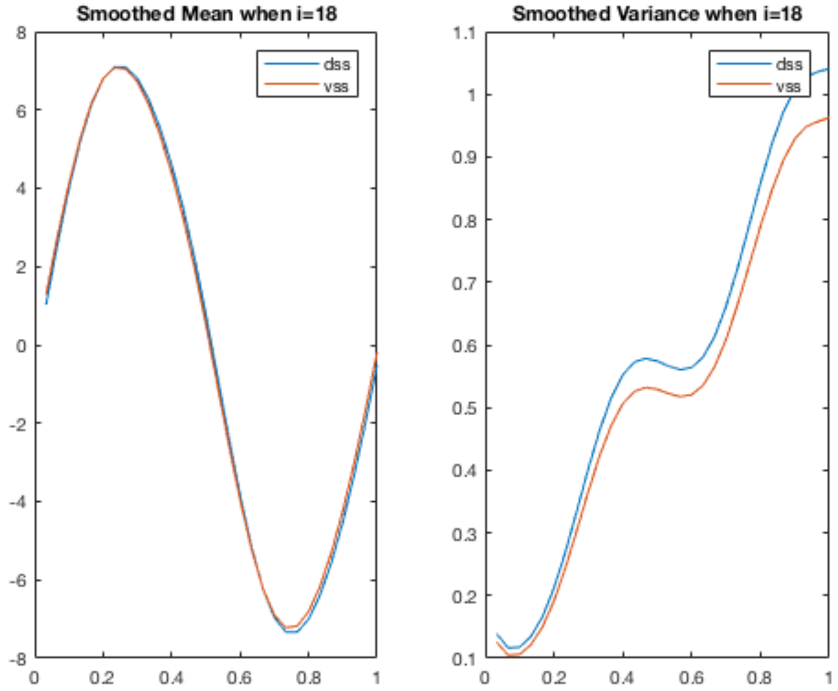

Smoothed Mean when i=13
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Smoothed Variance when i=13
dss
vss
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

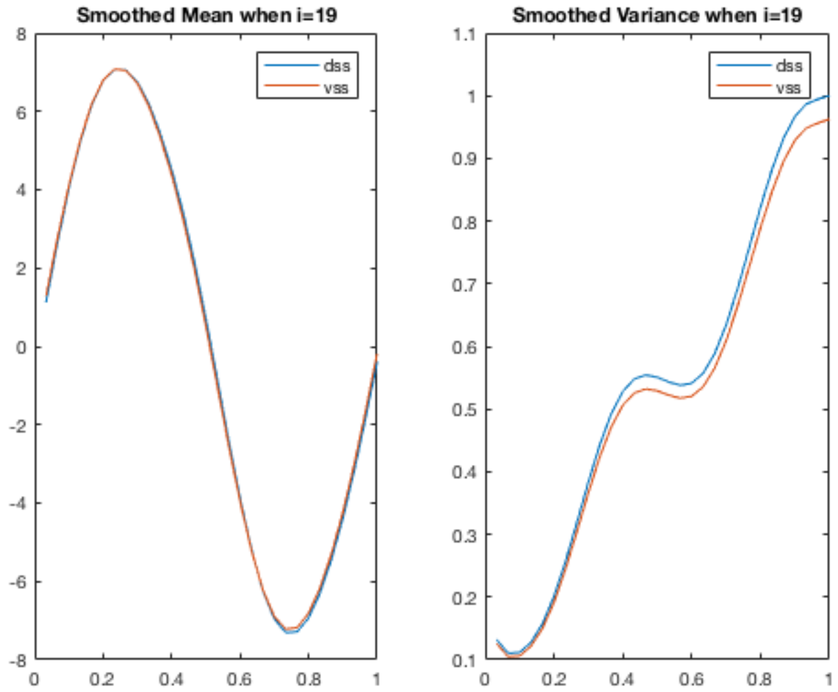

Smoothed Mean when i=14
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Smoothed Variance when i=14
dss
vss
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1

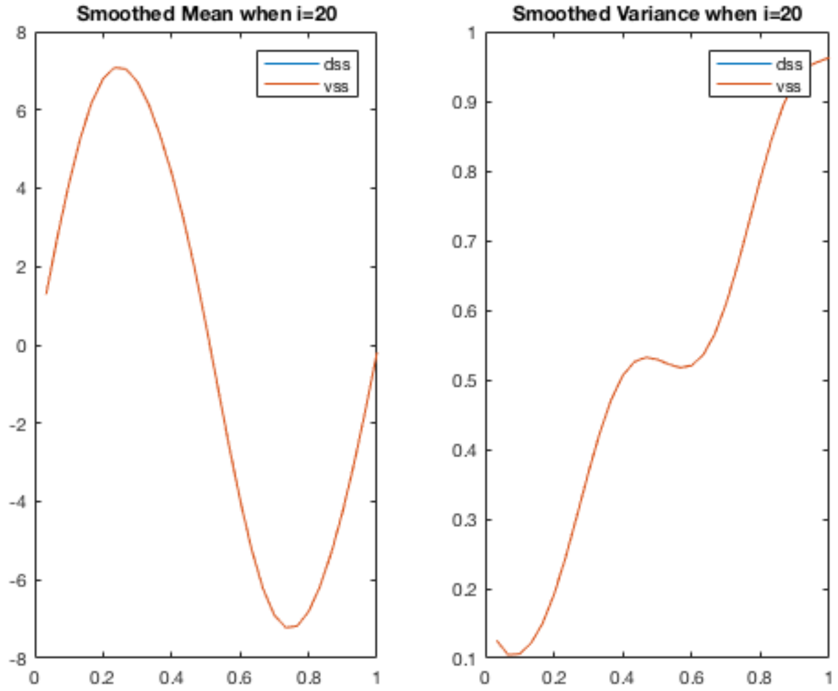

Smoothed Mean when i=15
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Smoothed Variance when i=15
dss
vss
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


Smoothed Mean when i=16
dss
vss
8
6
4
2
0
-2
-4
-6
-8
0
0.2
0.4
0.6
0.8
1
Smoothed Variance when i=16
dss
vss
1.2
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1


Smoothed Mean when i=17
Smoothed Variance when i=17
dss
vss
8
6
4
2
0
-2
-4
-6
-8
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
1


Smoothed Mean when i=18
Smoothed Variance when i=18
dss
vss
8
6
4
2
0
-2
-4
-6
-8
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
1





Estimation result

```
fprintf('-DSS estimate is %i \n', logparahat_dss(1))
fprintf('and the minimized objective value is %i. \n', val_dss);
fprintf('-VSS estimate is %i \n', logparahat_vss(1))
fprintf('and the minimized objective value is %i. \n', val_vss);
```

```
-DSS estimate is 2.045861e-01
and the minimized objective value is 1.117601e+03.
-VSS estimate is 2.045866e-01
and the minimized objective value is 1.117601e+03.
```

Published with MATLAB® R2017a