# Plotting descent

**Table of Contents**



## Load data

```
load("..\Data\Schlogl_1_2_descend_1_2straight_500_11-Jul-2022.mat")
```

```
%load("..\Data\Schlogl_1_2_descend_1_2straight_2000_11-Jul-2022.mat")
```

### Initialization values

```
eps_ic
```

```
eps_ic = 1
```

```
eps_min
```

```
eps_min = 1.0000e-03
```

```
err_thresh
```

```
err_thresh = 0.1000
```

```
f0_init
```

```
f0_init = 4
```

```
eps_S_thresh
```

```
eps_S_thresh = 1.0000e-07
```

```
delta_S_thresh
```

```
delta_S_thresh = 5.0000e-08
```

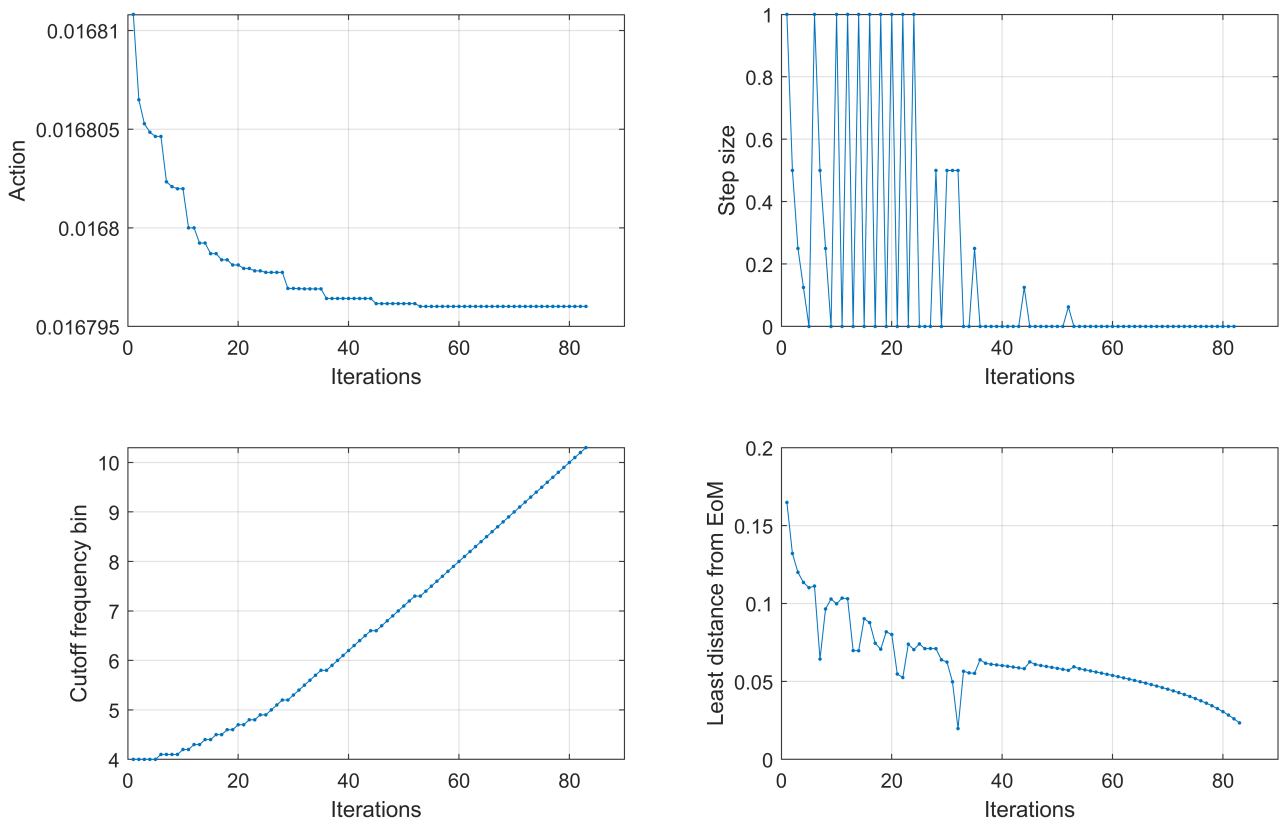
### Random code snippets for debugging

```
i=83;
deltax = picker_arr_idx(deltax_arr,traj_pt_arr,i);
deltax_s = picker_arr_idx(deltax_lp_arr,traj_pt_arr,i);
f0 = f0_arr(i);
save_plot_name = save_plot_name_root + '_deltax_'+string(i)+'.png';

c_out = debugging_deltax_plots(deltax,deltax_s,f0,plotnam,save_plot_name,1)
```

## Descent summary

```
[traj_pts,num_spec] = size(traj);

init_iter = 1;
tot_iter = size(S_arr,1);

%Action
figure()
subplot(2,2,1)
plot(S_arr,'.-','LineWidth',0.5)
ylabel('Action')
xlabel('Iterations')
grid on
xlim([0 ceil(size(S_arr,1)/10)*10])

%title(plotnam + ': Descent summary')

%epsilon-step size
subplot(2,2,2)
plot(eps_arr,'.-')
ylabel('Step size')
xlabel('Iterations')
grid on
xlim([0 ceil(size(S_arr,1)/10)*10])

%title(plotnam + ': Descent summary')

%cutoff frequency bin
subplot(2,2,3)
plot(f0_arr,'.-')
ylabel('Cutoff frequency bin')
xlabel('Iterations')
grid on
xlim([0 ceil(size(S_arr,1)/10)*10])


%Ham EoM distance frequency bin
subplot(2,2,4)
plot(a_arr,'.-')
ylabel('Least distance from EoM')
xlabel('Iterations')
grid on
xlim([0 ceil(size(S_arr,1)/10)*10])
```

```
x0=0.1;
y0=0.1;
width=26;
height=16;
set(gcf,'position',[x0,y0,width,height])
set(gcf,'units','centimeters','position',[x0,y0,width,height])
```

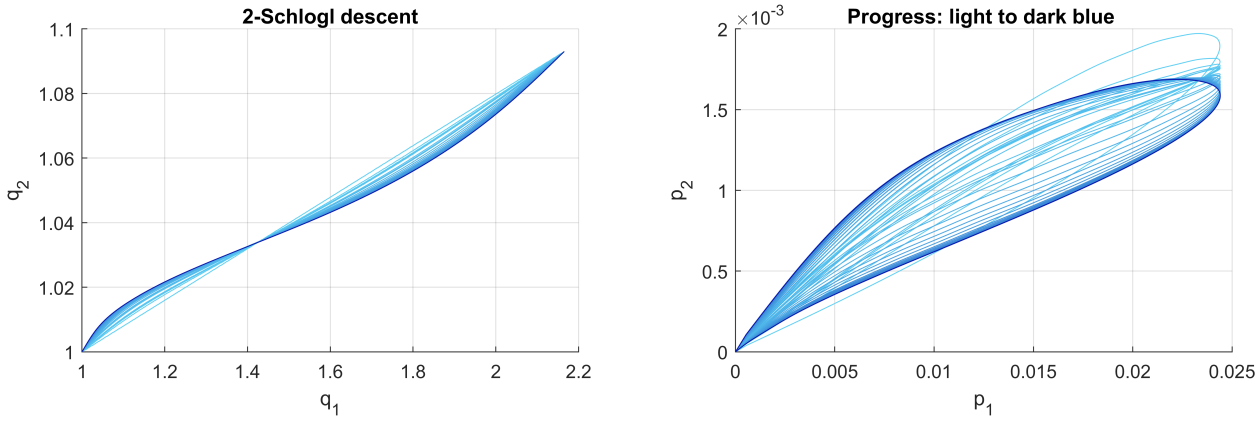



```
save_plot_name = save_plot_name_root + '_action_eps_freq_delta.png';
saveas(gcf,save_plot_name)
```

## Progress in phase space

```
%Trajectories
lightBLUE = [0.356862745098039,0.811764705882353,0.956862745098039];
darkBLUE = [0.0196078431372549,0.0745098039215686,0.670588235294118];
blueGRADIENTflexible = @(i,N) lightBLUE + (darkBLUE-lightBLUE)*((i-1)/(N-1));
diter  = 1;
figure()
subplot(1,2,1)
hold on
for i = init_iter:diter:tot_iter
    PS_traj = picker_arr_idx(PS_arr,traj_pt_arr,i);
    traj = PS_traj(:,1:num_spec);
    plot(traj(:,1),traj(:,2:end),'color',blueGRADIENTflexible(i,tot_iter))
end
```

```
xlabel("q_1")
ylabel("q_2")
title(plotnam+' descent')
hold off
grid on

%momentum
subplot(1,2,2)
hold on
for i = init_iter:diter:tot_iter
    PS_traj = picker_arr_idx(PS_arr,traj_pt_arr,i);
    traj = PS_traj(:,num_spec + 1:2*num_spec);
    plot(traj(:,1),traj(:,2:end),'color',blueGRADIENTflexible(i,tot_iter))
end
xlabel("p_1")
ylabel("p_2")
title("Progress: light to dark blue ")
hold off
grid on
x0=0.1;
y0=0.1;
width=28;
height=8;
set(gcf,'position',[x0,y0,width,height])
set(gcf,'units','centimeters','position',[x0,y0,width,height])
```



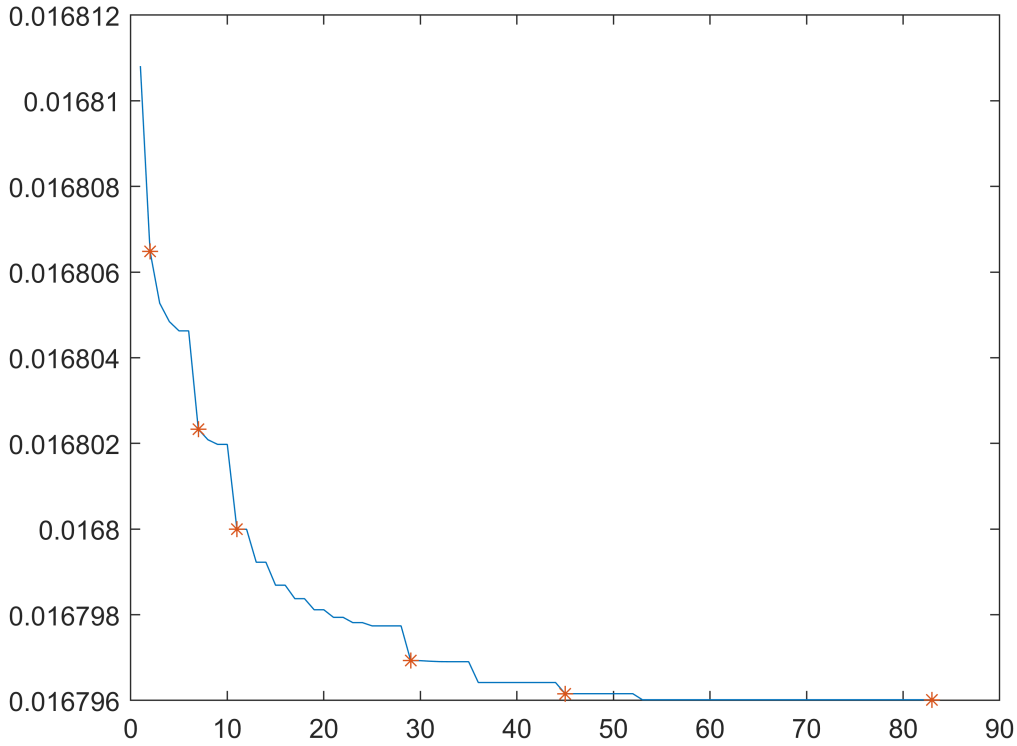

```
save_plot_name = save_plot_name_root + '_descender_progress.png';
saveas(gcf,save_plot_name)
```

## Plot sudden jump gradients

### Find spots

```
%Find index of sudden jumps of S_arr
%[TF,S1] = ischange(S_arr,'variance','MaxNumChanges',10);
TF = findchangepts(S_arr','MaxNumChanges',4);
 TF = [TF 45 size(S_arr,1)];
```

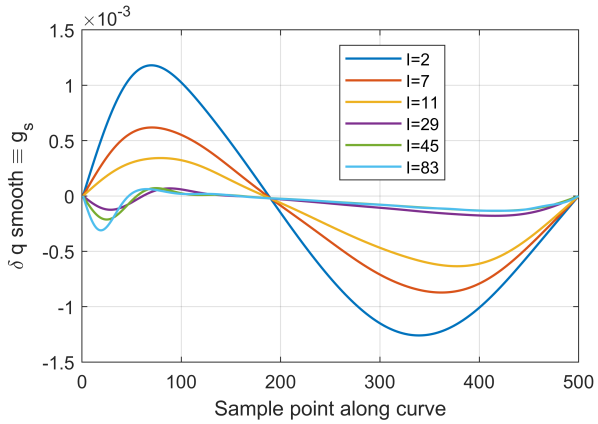
```
figure()
plot(S_arr)
hold on
xran = 1:size(S_arr,1);
plot(xran(TF),S_arr(TF),'*')
hold off
```

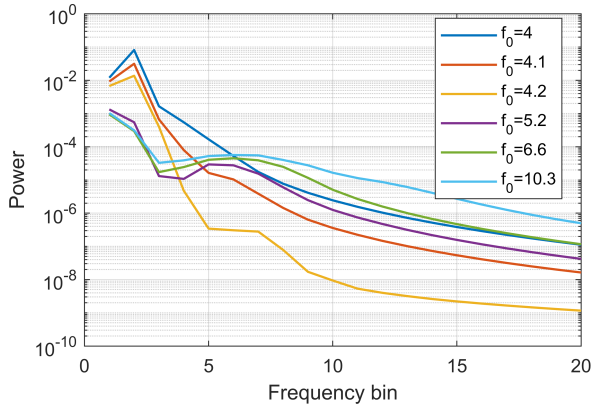



## Plot gradient

```
figure()
subplot(1,2,1)
for i = 1:size(TF,2)
    iter = TF(i);
    g_s = picker_arr_idx(deltax_lp_arr,traj_pt_arr,iter);
    plot(g_s(:,2),'DisplayName','I='+string(iter),'LineWidth',1.2)
    hold on
end
grid on
xlabel('Sample point along curve')
ylabel('\delta q smooth \equiv g_s')
legend('location','best')
%title(plotnam + ' : dim #2')
hold off
xlim([0 500])

subplot(1,2,2)
for i = 1:size(TF,2)
```

```
    iter = TF(i);
    g_s = picker_arr_idx(deltax_lp_arr,traj_pt_arr,iter);
    semilogy(abs(fft(g_s(:,2))).^2,'DisplayName','f_0='+string(f0_arr(iter)),'LineWidth',1.2)
    xlim([0 20])
    hold on
end
grid on
%title('Descent progress')
xlabel('Frequency bin')
ylabel('Power')
legend('location','best')
hold off


x0=0.1;
y0=0.1;
width=28;
height=8;
set(gcf,'position',[x0,y0,width,height])
set(gcf,'units','centimeters','position',[x0,y0,width,height])
```



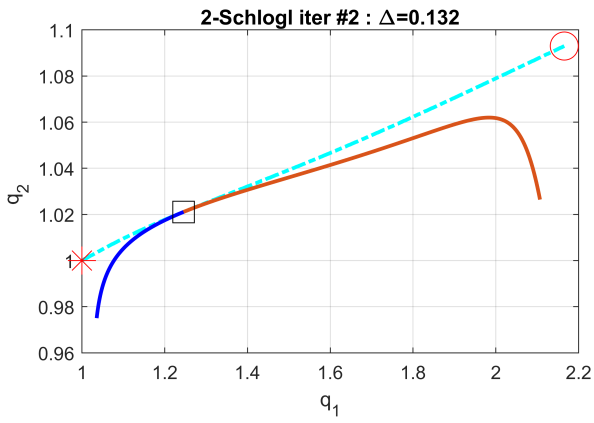

```
save_plot_name = save_plot_name_root + '_g_s_change_2.png';

saveas(gcf,save_plot_name)
```

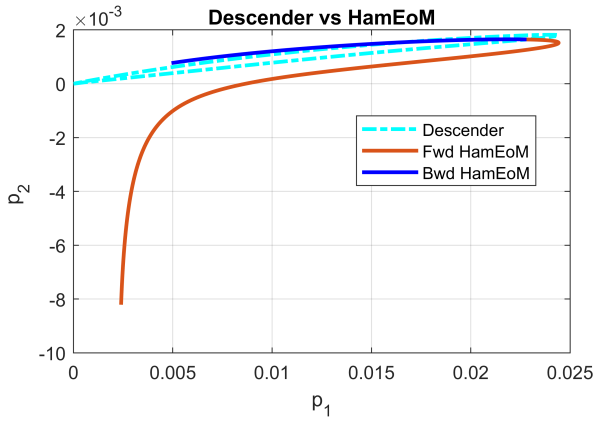
```
plotnam_old = plotnam;
```

## Hamilton's EoM

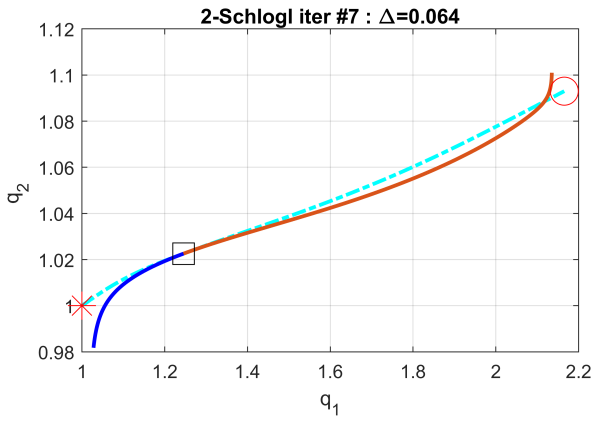
```
for i = 1:size(TF,2)
    iter = TF(i);
    PS_traj = picker_arr_idx(PS_arr,traj_pt_arr,iter);
    t_traj = picker_arr_idx(time_arr,traj_pt_arr,iter);
    save_plot_name = save_plot_name_root + '_HamEoM_'+string(iter)+'.png';
    plotnam = plotnam_old + ' iter #'+string(iter);
    Ham_closest_approach(PS_traj,S_traj,t_traj,...
    dHamdp_fun, dHamdq_fun, save_plot_name,1,plotnam)
```

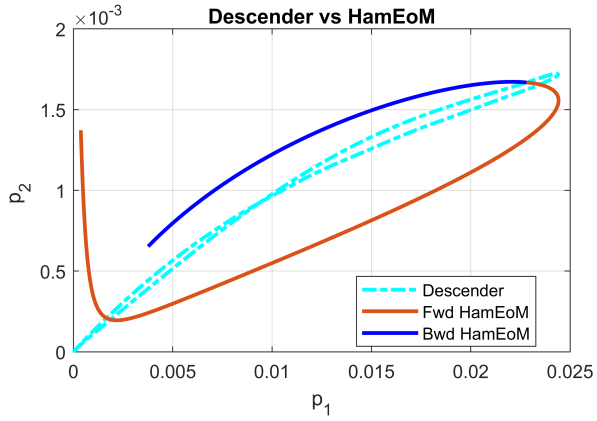
```
end
```



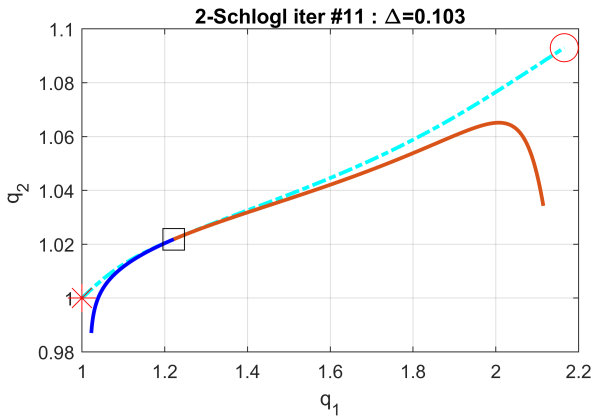



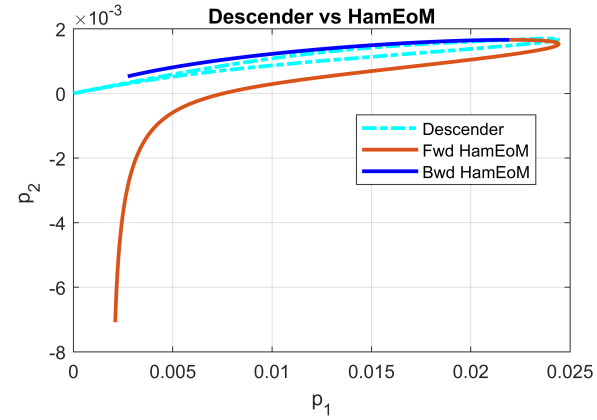

```
ans = 0.1321
```



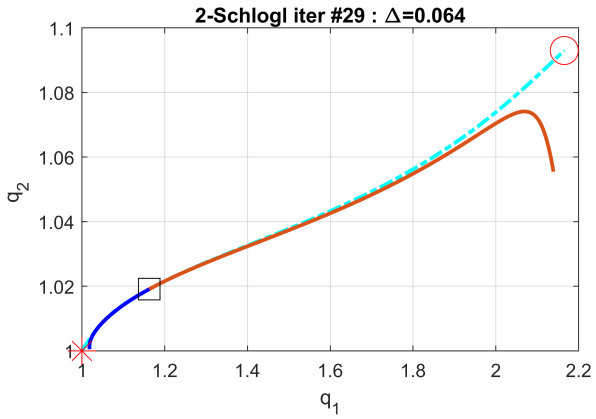



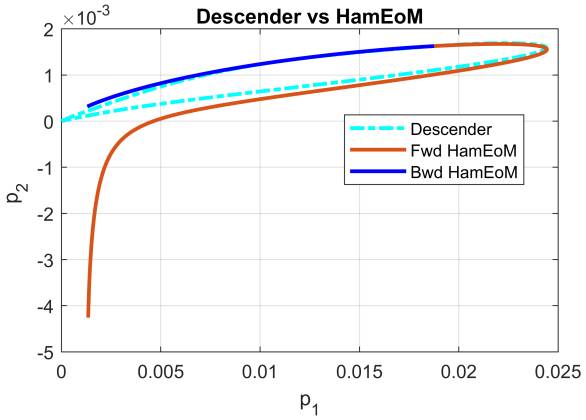

```
ans = 0.0643
```



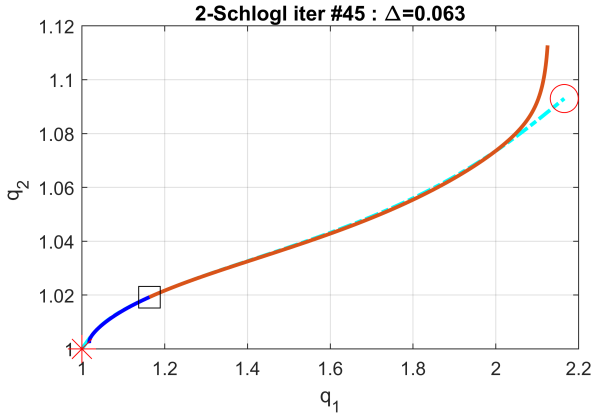



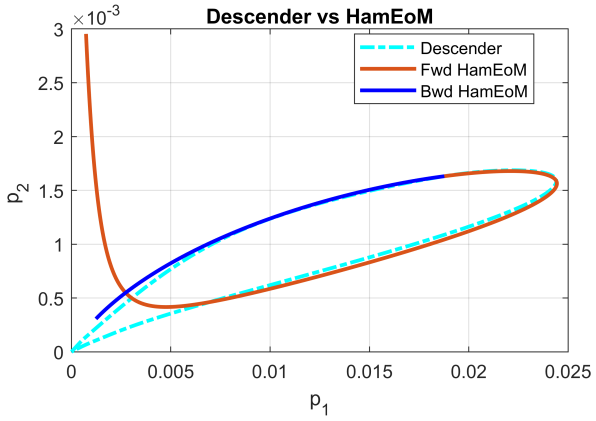

```
ans = 0.1035
```



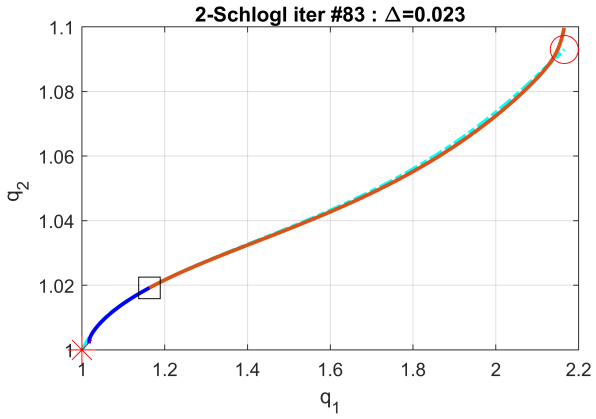



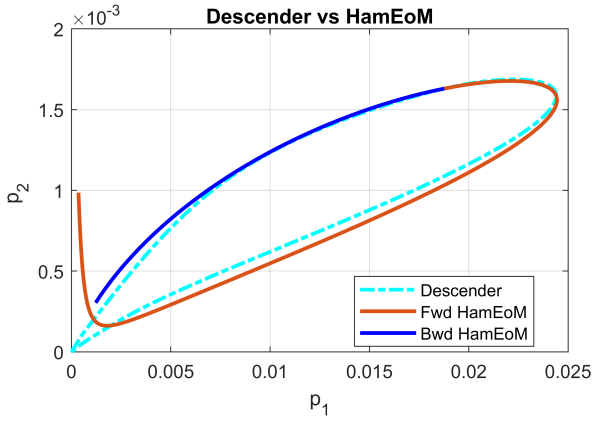

```
ans = 0.0638
```




```
ans = 0.0625
```



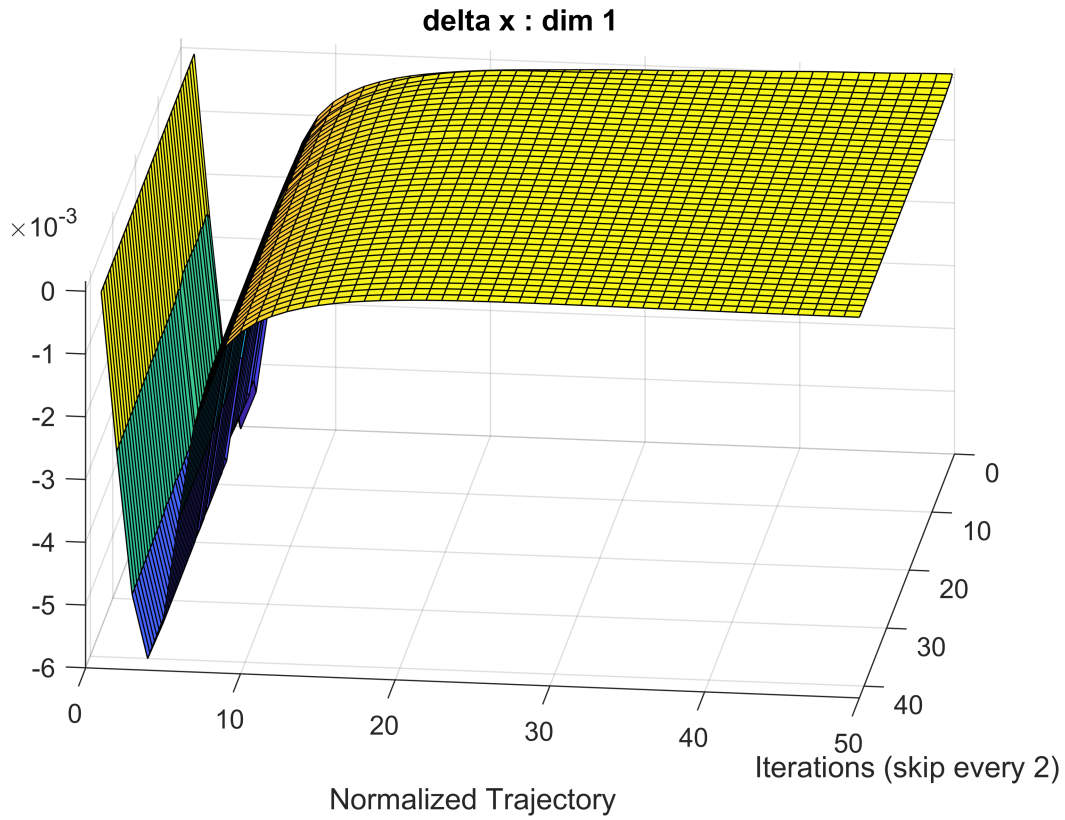

```
ans = 0.0234
```

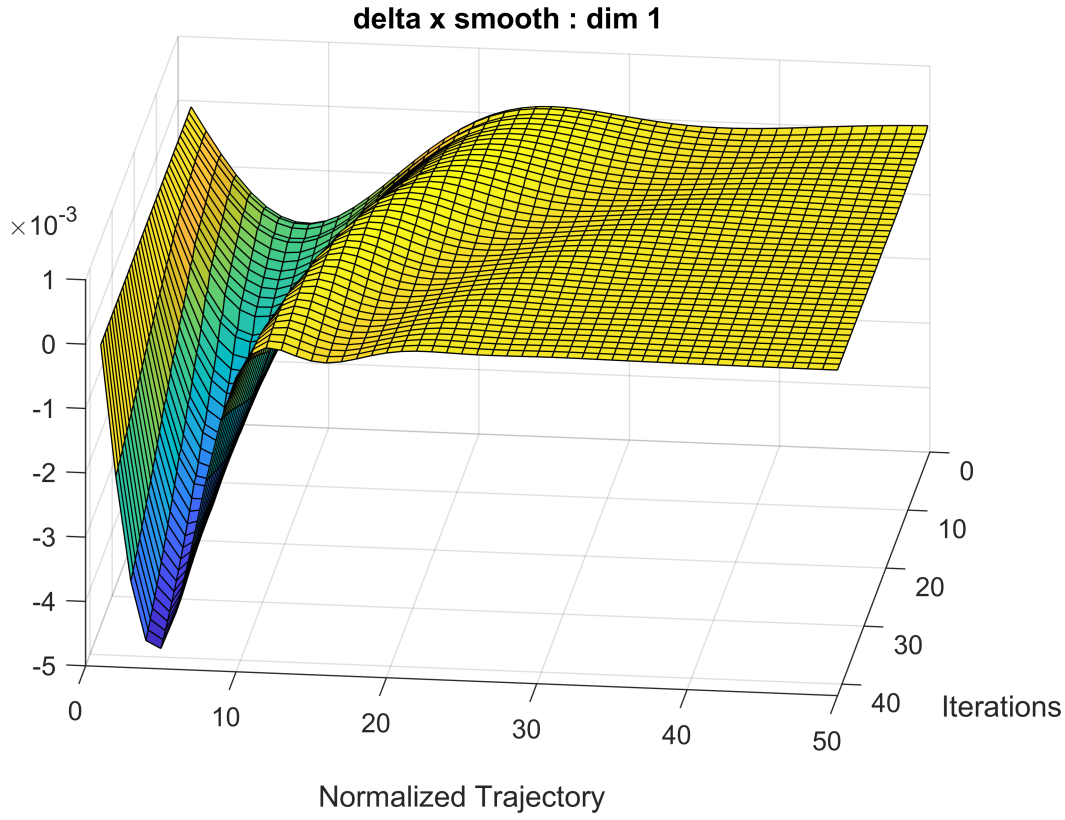
```
plotnam = plotnam_old;
```

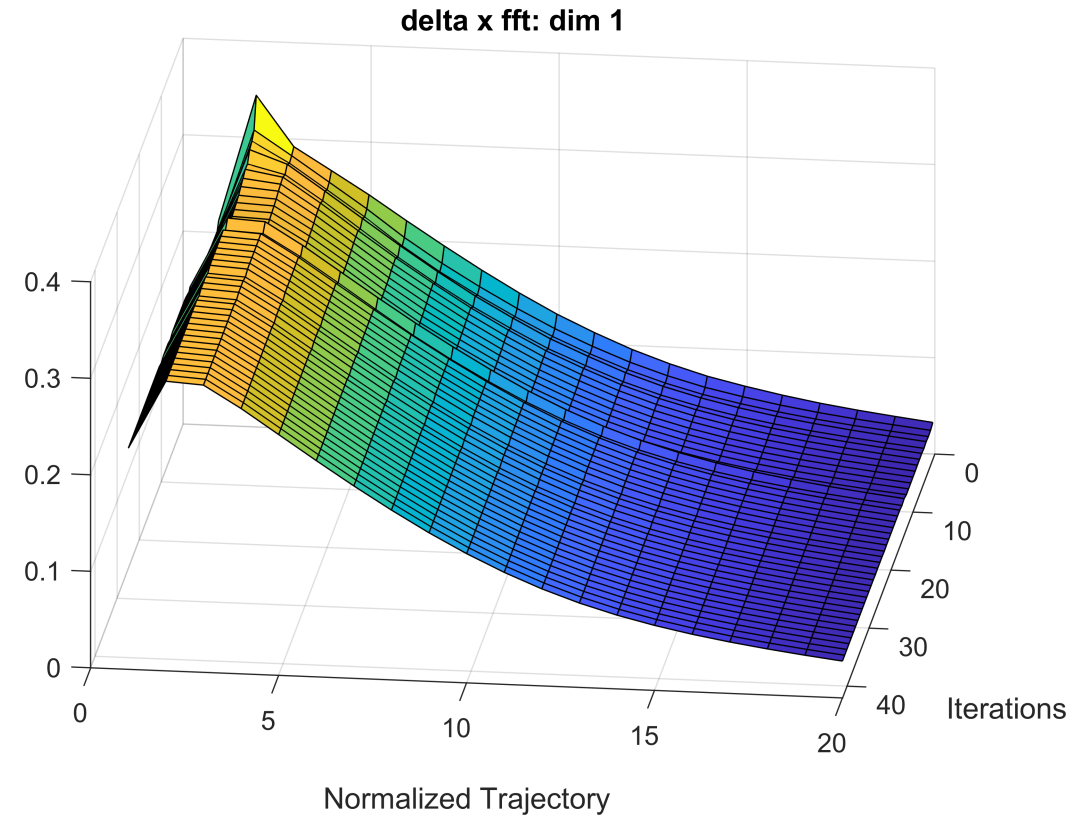
## Surf plots

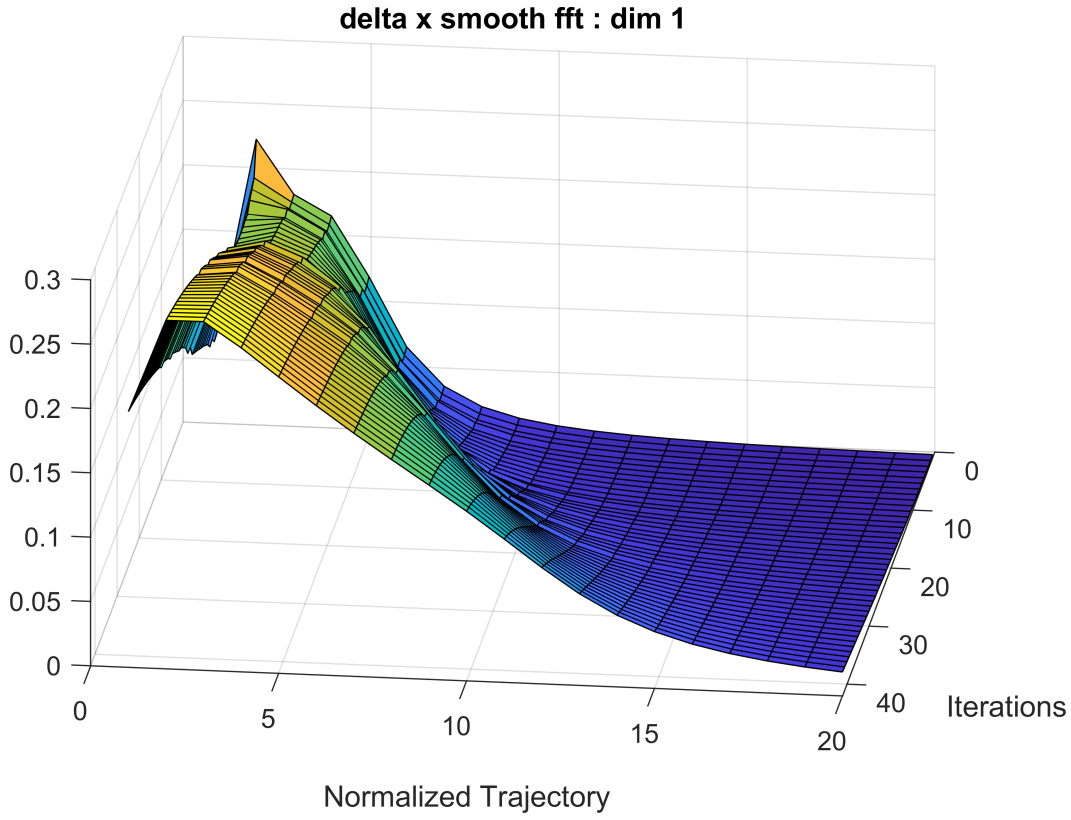
```
%Surf plots
%Points to skip along trajectory
dtraj = 10;
diter = 2;

traj_pts_indx = 1000;
tot_iter = 83;
init_iter = 1;
```

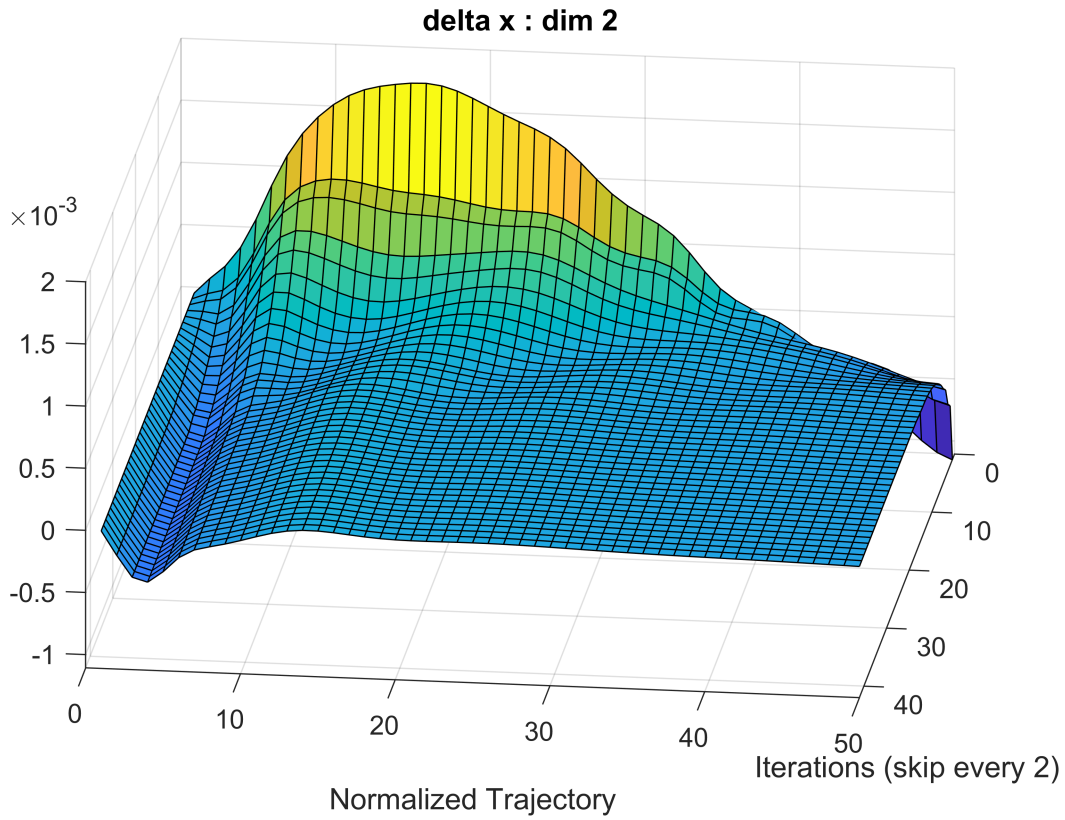
```
%deltax
[deltax_3d, deltax_3d_fft] = surf_input(deltax_arr,num_spec,tot_iter,traj_pt_arr,traj_pts_indx)
[deltax_s_3d, deltax_s_3d_fft] = surf_input(deltax_lp_arr,num_spec,tot_iter,traj_pt_arr,traj_pt

for i = 1:num_spec
    figure()
    surf(squeeze(deltax_3d(i,1:dtraj:traj_pts,init_iter:diter:tot_iter)))
    %shading interp
    title('delta x : dim '+string(i))
    view([97.02 32.46])
    xlabel('Iterations (skip every '+string(diter)+')')
    ylabel('Normalized Trajectory')

    figure()
    surf(squeeze(deltax_s_3d(i,1:dtraj:traj_pts,init_iter:diter:tot_iter)))
    %shading interp
    title('delta x smooth : dim '+string(i))
    view([97.02 32.46])
    xlabel('Iterations')
    ylabel('Normalized Trajectory')

    figure()
    surf(squeeze(deltax_3d_fft(i,1:20,init_iter:diter:tot_iter)))
    %shading interp
    title('delta x fft: dim '+string(i))
    view([97.02 32.46])
    xlabel('Iterations')
    ylabel('Normalized Trajectory')

    figure()
    surf(squeeze(deltax_s_3d_fft(i,1:20,init_iter:diter:tot_iter)))
    %shading interp
    title('delta x smooth fft : dim '+string(i))
    view([97.02 32.46])
    xlabel('Iterations')
    ylabel('Normalized Trajectory')

end
```

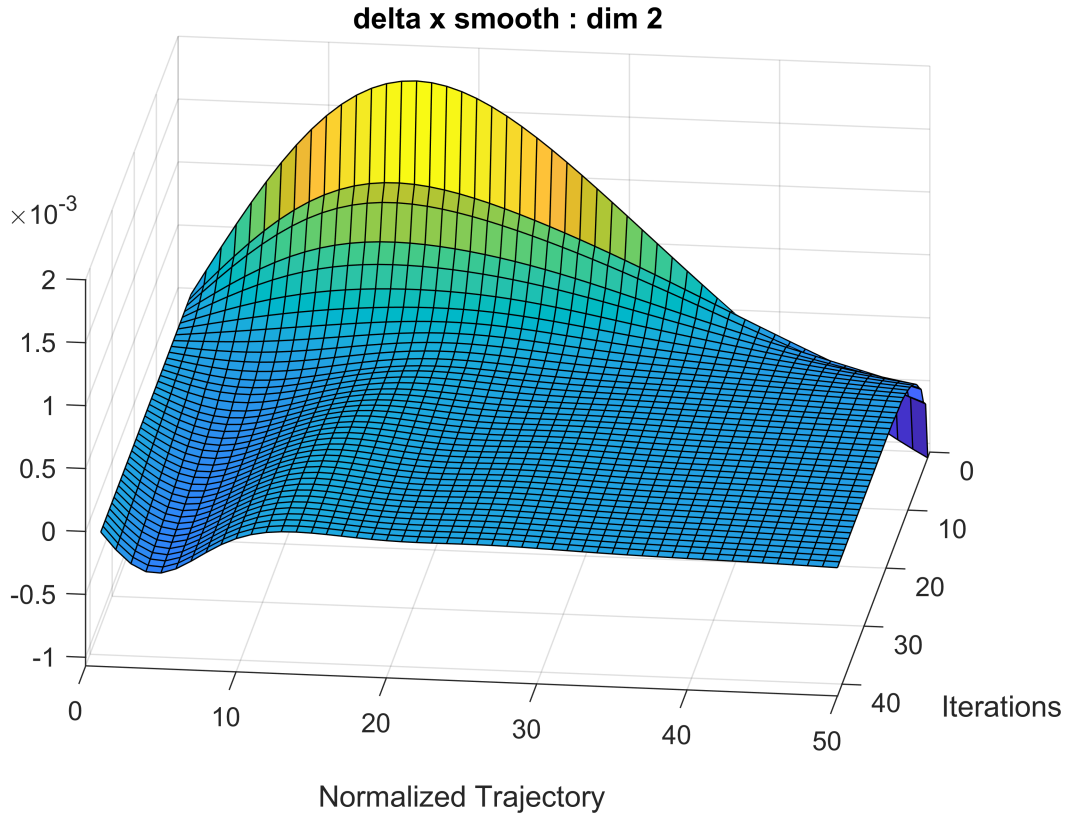

delta x : dim 1
×10-3
0
-1
-2
-3
-4
-5
-6
0
10
20
30
40
50
Normalized Trajectory
0
10
20
30
40
Iterations (skip every 2)

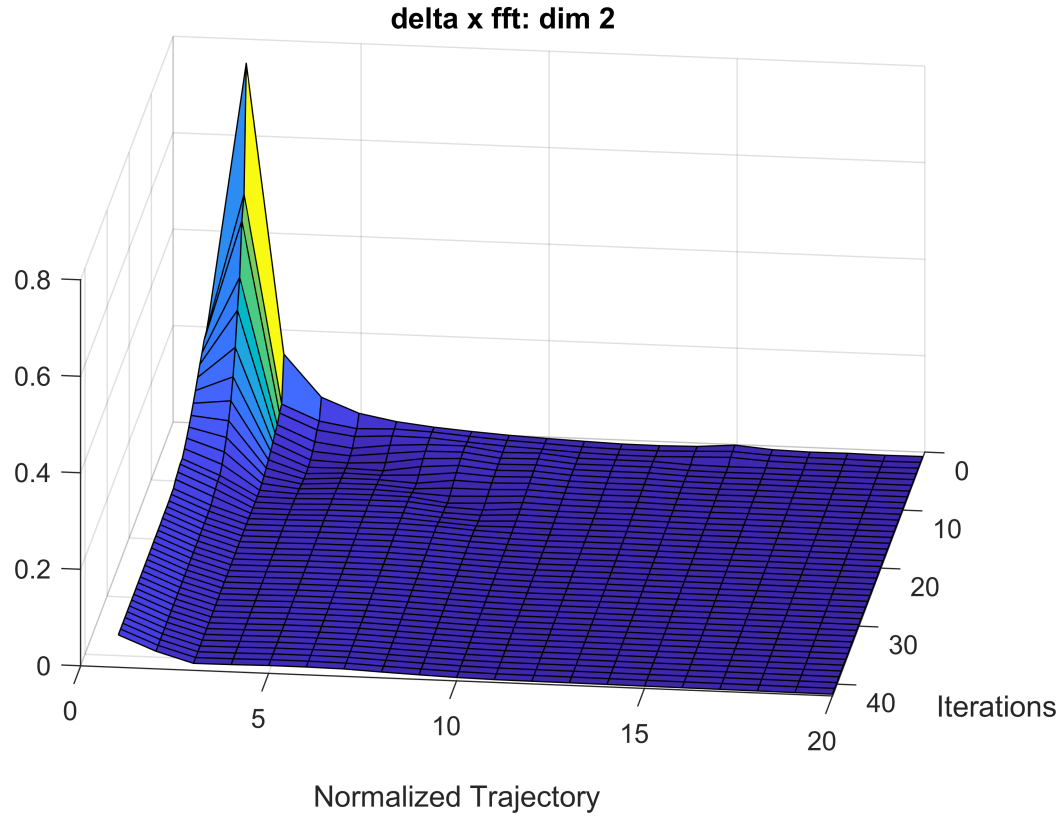

delta x smooth : dim 1
×10-3
1
0
-1
-2
-3
-4
-5
0
10
20
30
40
50
Normalized Trajectory
0
10
20
30
40
Iterations

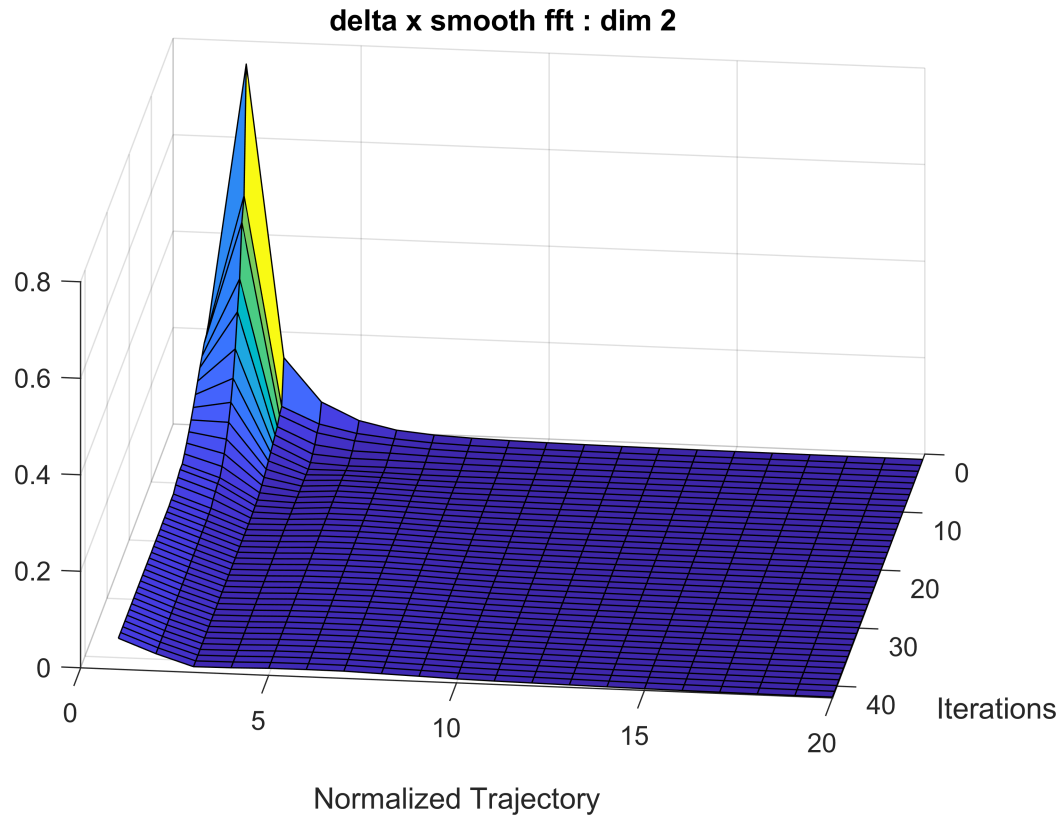

delta x fft: dim 1
0.4
0.3
0.2
0.1
0
0
5
10
15
20
Normalized Trajectory
0
10
20
30
40
Iterations


delta x smooth fft : dim 1
0.3
0.25
0.2
0.15
0.1
0.05
0
0
5
10
15
20
Normalized Trajectory
0
10
20
30
40
Iterations


delta x : dim 2
×10-3
2
1.5
1
0.5
0
-0.5
-1
0
10
20
30
40
50
Normalized Trajectory
0
10
20
30
40
Iterations (skip every 2)

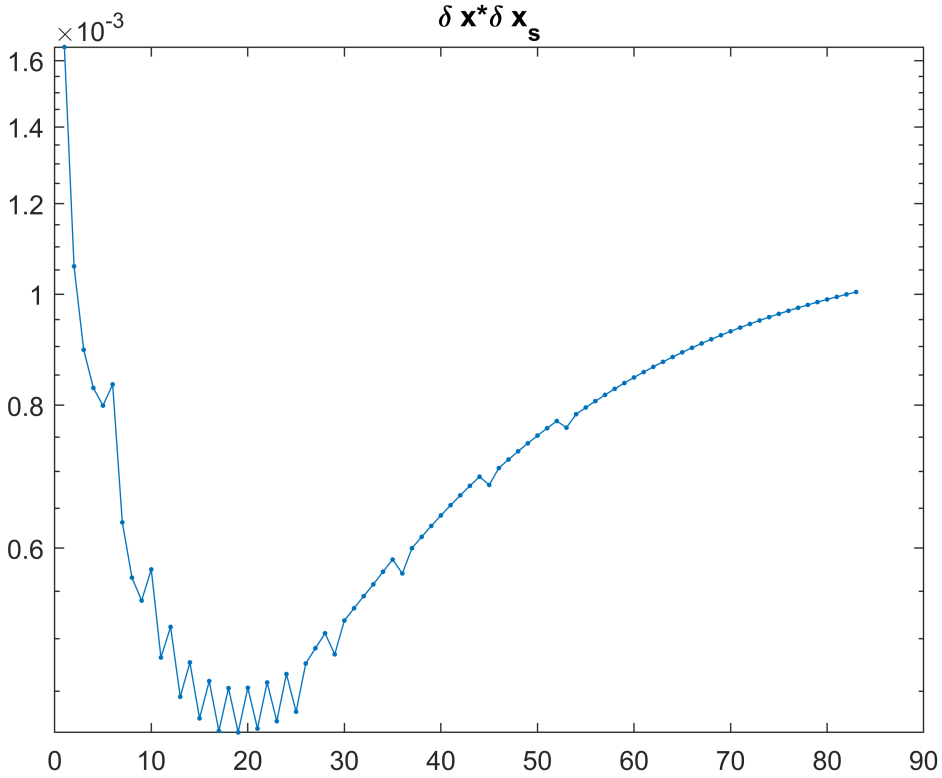

delta x smooth : dim 2
×10-3
2
1.5
1
0.5
0
-0.5
-1
0
10
20
30
40
50
Normalized Trajectory
0
10
20
30
40
Iterations



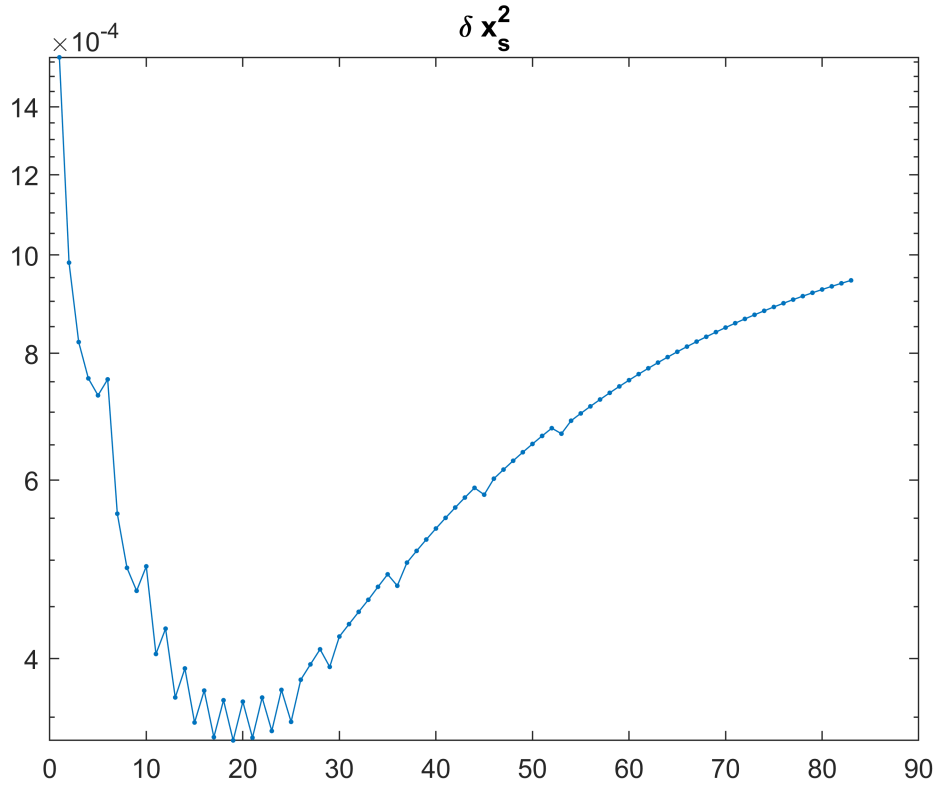




```
%Plot delta_x^2
delta_S = zeros(tot_iter,1);
```

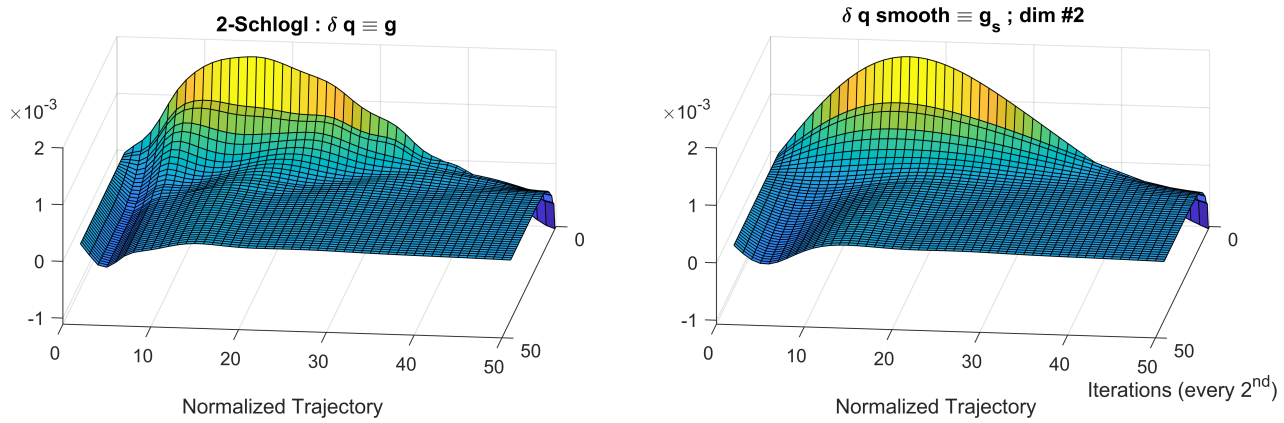
```
for i = 1:tot_iter
    delta_S(i) = sum(squeeze(deltax_3d(:,:,i)).*squeeze(deltax_s_3d(:,:,i)),"all");
end
figure()
semilogy(delta_S,'.-')
title('\delta x*\delta x_s')
```




```
%Plot delta_x_s^2
delta_S = zeros(tot_iter,1);
for i = 1:tot_iter
    delta_S(i) = sum(squeeze(deltax_s_3d(:,:,i)).*squeeze(deltax_s_3d(:,:,i)),"all");
end
figure()
semilogy(delta_S,'.-')
title('\delta x_s^2')
```




```
i=2;
figure()
subplot(1,2,1)
surf(squeeze(deltax_3d(i,1:dtraj:traj_pts,init_iter:diter:tot_iter)))
%shading interp
title(plotnam + ' : \delta q \equiv g ')
view([97.02 32.46])
ylabel('Normalized Trajectory')

subplot(1,2,2)
surf(squeeze(deltax_s_3d(i,1:dtraj:traj_pts,init_iter:diter:tot_iter)))
%shading interp
title('\delta q smooth \equiv g_s ; dim #'+string(i))
view([97.02 32.46])
xlabel('Iterations (every '+string(diter)+'^{nd})')
ylabel('Normalized Trajectory')


x0=0.1;
y0=0.1;
width=28;
height=8;
set(gcf,'position',[x0,y0,width,height])
set(gcf,'units','centimeters','position',[x0,y0,width,height])
```




```
save_plot_name = save_plot_name_root + '_delta_surf_2.png';

saveas(gcf,save_plot_name)
```

## Old stuff

```
load("..\Data\Schlogl_1_2_descend_1_2straight_2000_11-Jul-2022.mat")
a_arr = [];
max_iter = size(S_arr,1);
```

```
for iter = 1:max_iter
    PS_traj  = picker_arr_idx(PS_arr,traj_pt_arr,iter);
    S_traj = picker_arr_idx(S_traj_arr,traj_pt_arr,iter);
    t_traj = picker_arr_idx(time_arr,traj_pt_arr,iter);

    a_min = Ham_closest_approach_noplot(PS_traj,S_traj,t_traj,dHamdp_fun, dHamdq_fun,save_plot_
    a_arr = [a_arr; a_min];
end
```

```
iter = 56;
PS_traj  = picker_arr_idx(PS_arr,traj_pt_arr,iter);
S_traj = picker_arr_idx(S_traj_arr,traj_pt_arr,iter);
t_traj = picker_arr_idx(time_arr,traj_pt_arr,iter);

a_min = Ham_closest_approach(PS_traj,S_traj,t_traj,dHamdp_fun, dHamdq_fun,save_plot_name,1,plot
```

```
figure()
plot(a_arr)
grid on
```

```
figure()
```

```
plot(a_arr,'.-')
xlim([1 56])
grid on
```

```
save("..\Data\Schlogl_1_2_descend_1_2straight_2000_11-Jul-2022.mat")
```

```
figure()
plot(traj_pt_arr)
```

```
%Trajectories
lightBLUE = [0.356862745098039,0.811764705882353,0.956862745098039];
darkBLUE = [0.0196078431372549,0.0745098039215686,0.670588235294118];
blueGRADIENTflexible = @(i,N) lightBLUE + (darkBLUE-lightBLUE)*((i-1)/(N-1));
diter  = 1;
figure()
hold on
for i = init_iter:diter:tot_iter
    PS_traj = picker_arr_idx(PS_arr,traj_pt_arr,i);
    traj = PS_traj(:,1:num_spec);
    plot(traj(:,1),traj(:,2:end),'color',blueGRADIENTflexible(i,tot_iter))
end
xlabel("q_1")
ylabel("rest")
title("Descender progress: light blue to dark blue ")
hold off
grid on
```